INDEKS Nr 32034 X ISSN 1230-7947

Commodore

2/1994

AMIGOWIEC

Pismo użytkowników komputerów AMIGA

cena 25.000 zł

KOPROCESOR dla A1200

Oprogramowanie:

- ✓ BRILLIANCE
- ✓ OPUS 4.11

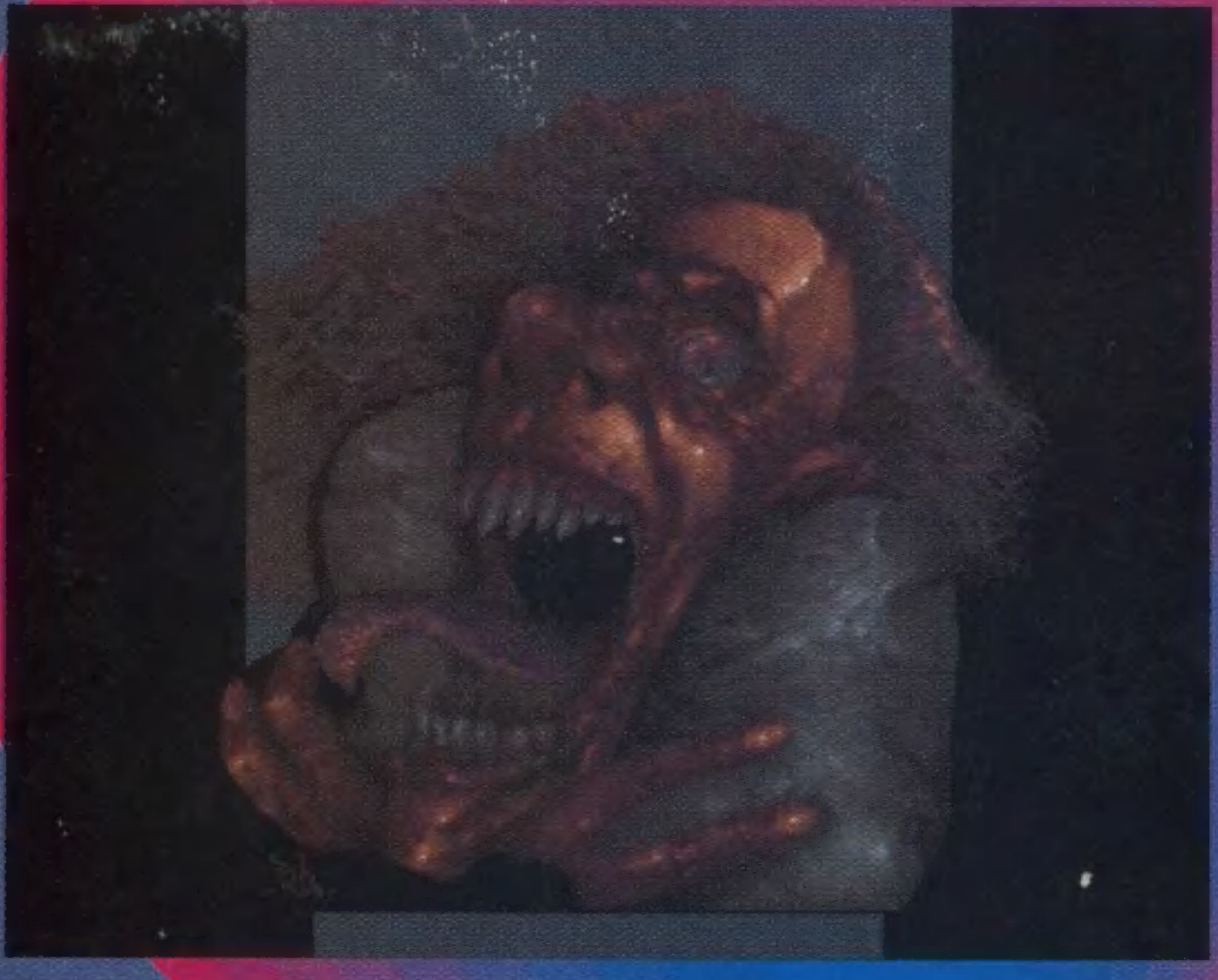
Sprzęt:

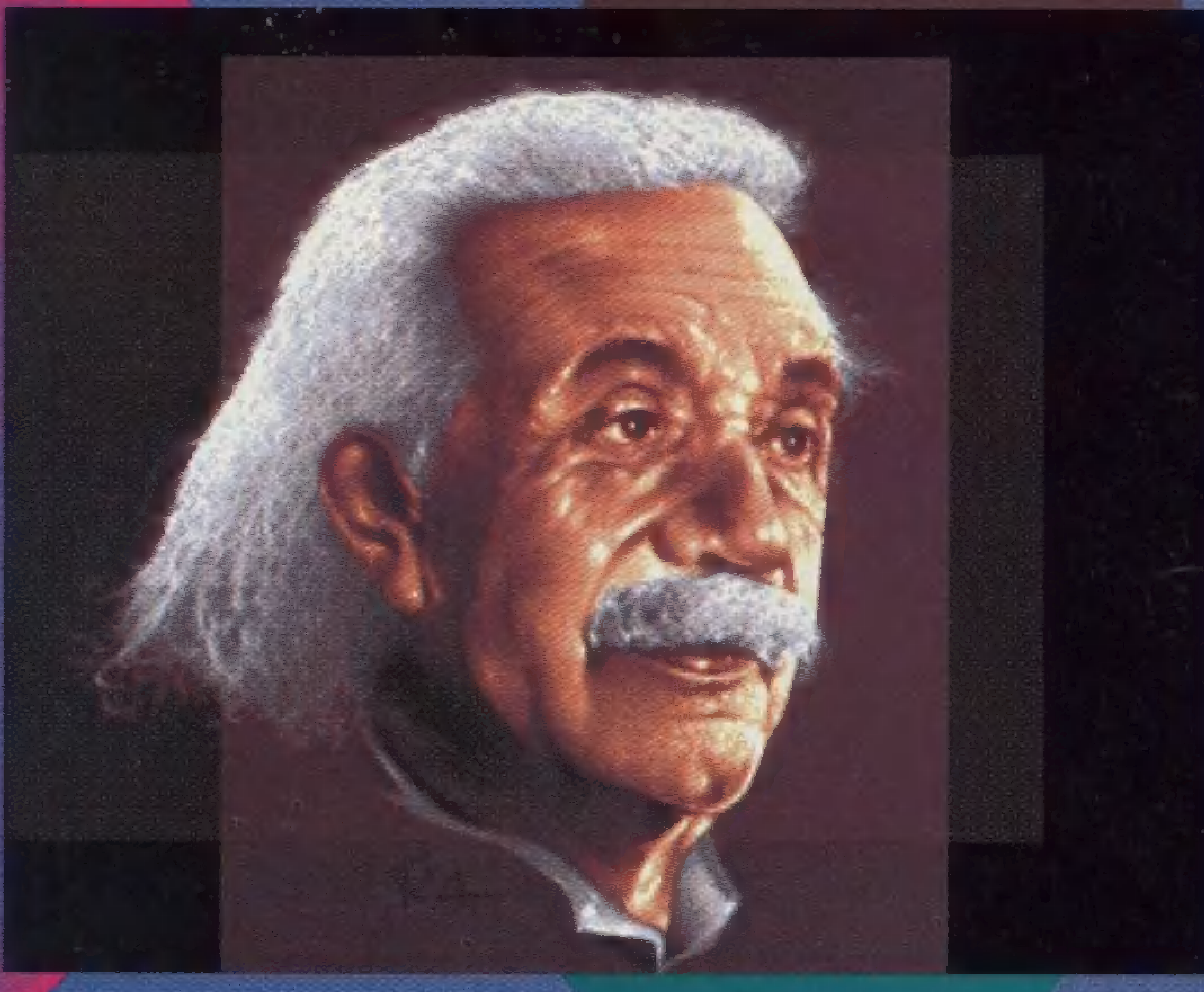
- ✓ AMTARI
- ✓ FASTLANE Z3

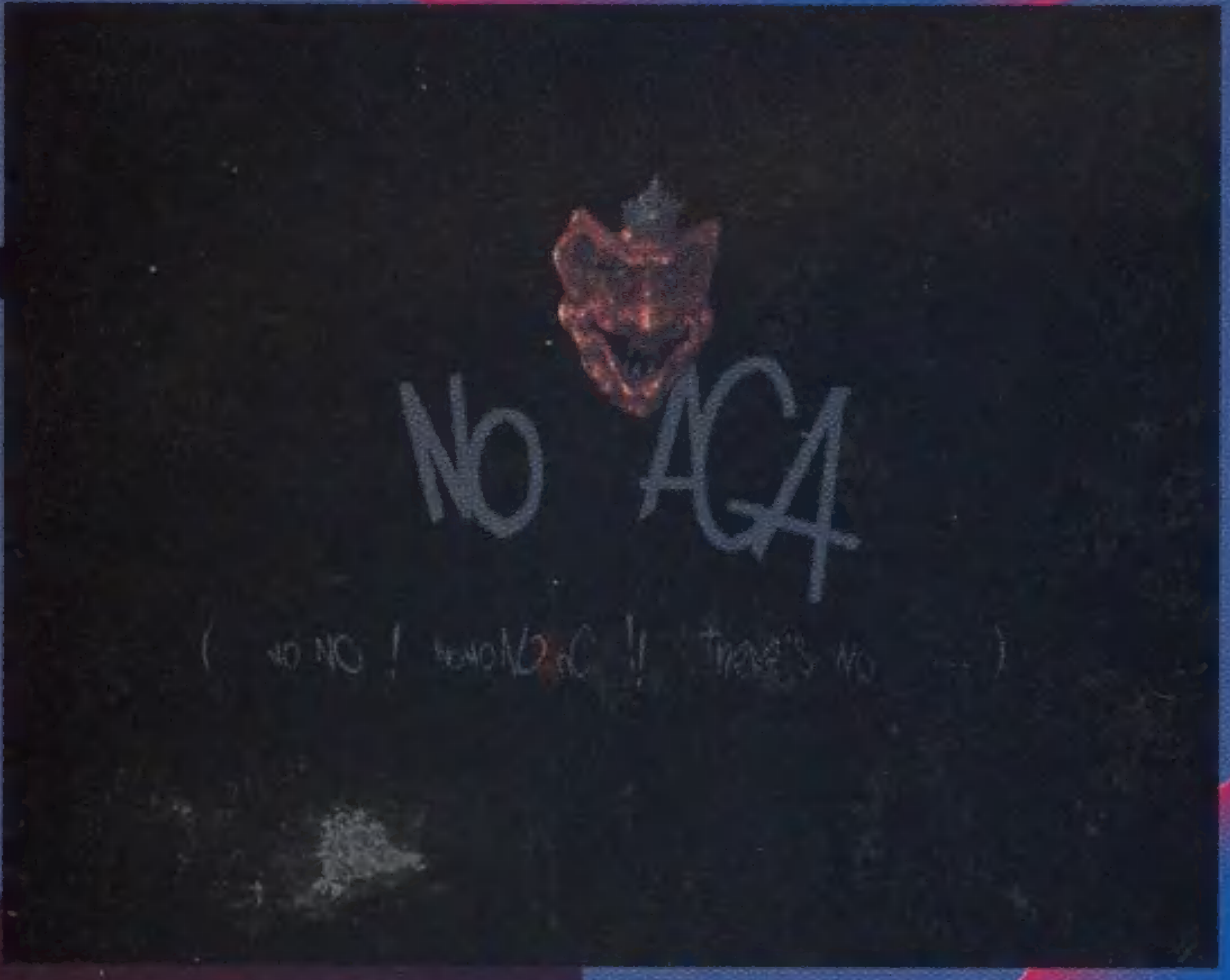
WARSZTATY!

- AMOS
- Studio
- Arexx cz. 3
- Imagine cz. 2
- Samplowanie
- Supra fax-modem 14400

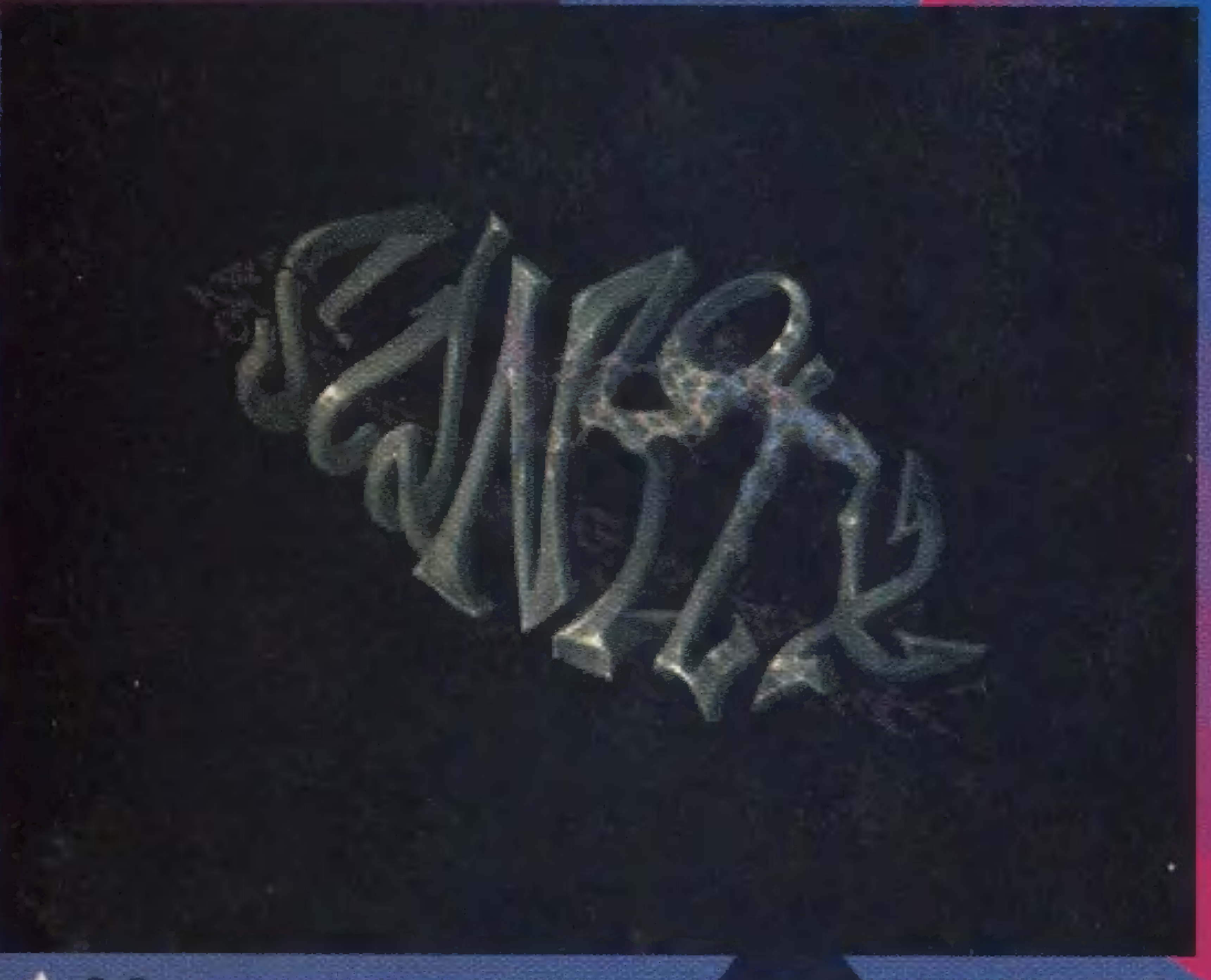
Ciąg dalszy na stronie 75

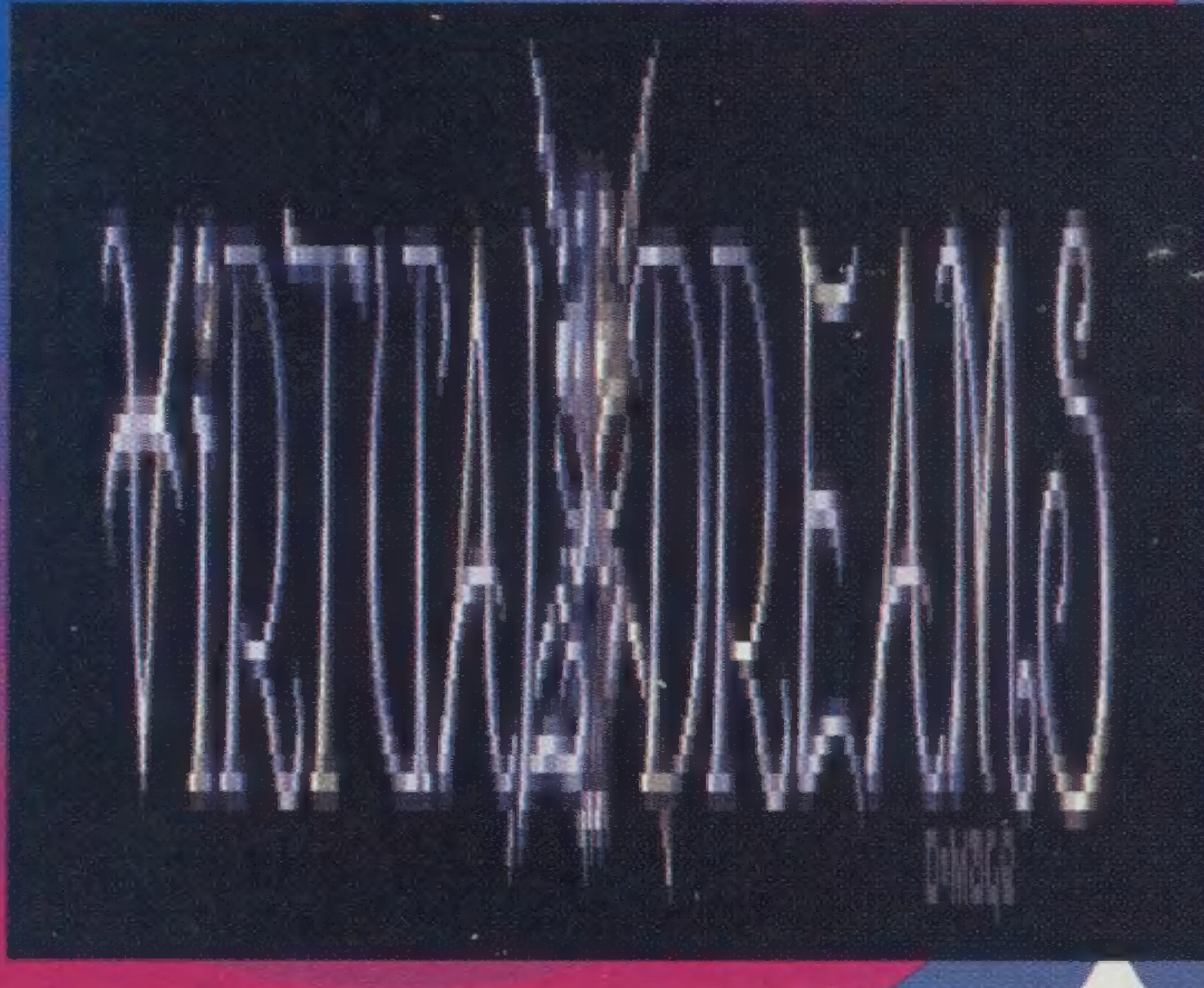





NO AGA






VIRTUAL DREAMS

AMIGOWIEC
ROK 5 NR 2/94 (36) INDEX 32034X

REDAKCJA

REDAKTOR NACZELNY:
RYSZARD KOWALSKI

Z-CA REDAKTORA NACZELNEGO:
KRZYSZTOF NOWICKI

SEKRETARZ REDAKCJI:
TOMASZ ŁOBODA

OPRACOWANIE GRAFICZNE:
KRZYSZTOF WIRSZYŁŁO,
BARTŁOMIEJ WRÓBLEWSKI

SKŁAD KOMPUTEROWY:
MAŁGORZATA LEWANDOWSKA

KOREKTA:
KAROL JAGODZIŃSKI,
ŁUCJA UMIŃSKA

KOLPORTAŻ:
ANDRZEJ KENTZER

PRENUMERATA:
AGNIESZKA PRZYBYLSKA

WSPÓŁPRACA:
WOJCIECH BIAŁKOWSKI,
WOJCIECH CZYŻ,
JAROSŁAW CHROSTOWSKI,
MARCIN GACKOWSKI,
ADAM GREGROWICZ,
TOMASZ FLANC,
TOMASZ HRYCUNIAK,
MACIEJ KLIMKIEWICZ,
TOMASZ KOKOSZCZYŃSKI,
SEBASTIAN KŁOMSKI,
TOMASZ KULBACKI,
MARIUSZ ŁAWICKI
TOMASZ ŁOBODA,
ARTUR ŁUKASIK,
TOMASZ MATAJEK,
ROBERT RUTKOWSKI,
IZABELA SKIBIŃSKA,
MAREK STOR,
KRZYSZTOF TROJANOWSKI,

ADRES REDAKCJI:
ul. Świętojańska 2/7
85-017 Bydgoszcz
tel. (+52) 28-79-20, fax (+52) 22-64-03

KONTO BANK PKO S.A. BYDGOSZCZ
00509011-04009339-2511-30-00111o

SKŁAD NA KOMPUTERACH AMIGA
OKŁADKA: WOJCIECH BIAŁKOWSKI
DRUK: "ZAKŁADY GRAFICZNE" SP. Z O.O.
UL. OKRZEI 5, 64-920 PIŁA

Ruszyło z kopyta...



Przed Wami kolejny numer naszego pisma. Ruszyło z kopyta - chciałoby się rzec, patrząc na amigowy rynek z perspektywy pierwszych miesięcy nowego roku. Zarówno w dziedzinie sprzętu, jak i oprogramowania została przerwana stagnacja trwająca od września 1994 roku. Warto by się zastanowić, dlaczego w ogóle taka stagnacja miała miejsce? Możemy tylko domniemywać, że główną przyczyną była bardzo niepewna sytuacja Commodore International pod koniec zeszłego roku. Brak kompetencji kierownictwa oraz związane z tym wyniki finansowe korporacji osiągnęły swoje apogeum, co zaowocowało daleko idącymi zmianami w samej firmie. Commodore zmienił kierownictwo, zredukował koszty i wreszcie postawił sprawę jasno: "naszym głównym produktem jest Amiga", kończąc tym samym okres wewnętrznego bałaganu i niejasnych posunięć. Całokształt tych zmian zaowocował na początku roku 1994 zdumiewającą ilością nowych produktów. Szczególnie firmy amerykańskie pokazały wreszcie lwi pazur. GVP zapowiedziało (wypuściło) nowe karty turbo do Amigi 1200 i 4000, a także kartę TBC. Centaur Development pokazało przez cały rok 1993 "ukrywane" moduły do Opal Vision. Naliczyłem ponad 5 nowych kart graficznych, sporo muzycznych i sieciowych, nie wspominając już o kontrolerach HD, czy wszelkiego rodzaju rozszerzeniach. Na polu oprogramowania rzecz ma się podobnie. Amiga za sprawą firm Soft Logic, Soft Wood oraz Palatinum Software zaatakowała zarezerwowaną dotychczas dla Maca i PC sferę DTP. Jak się okazało, nie bez powodzenia. Niezliczona ilość nowych wersji programów wspierających najróżniejsze karty graficzne oraz nowe kości AGA. Jak na okres 3 miesięcy bilans jest chyba niezwykle korzystny.

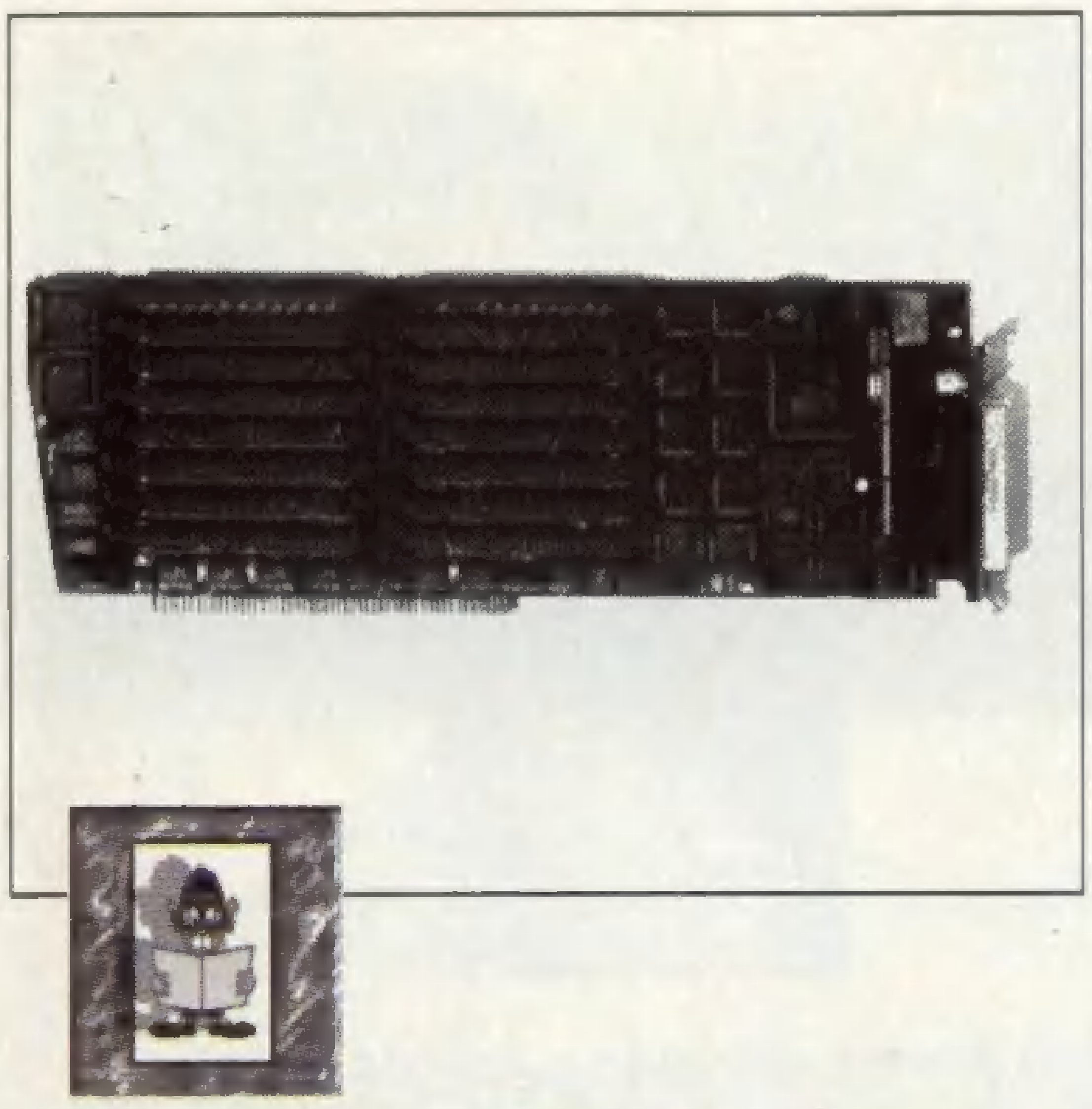
My tymczasem wróćmy do numeru bieżącego. Dzisiaj w Trybunie krótka polemika na temat sposobów porównywania Amigi i PC. Autor zwraca uwagę na prawidłowe kryteria porównania i odpowiedzialność wypowiedzi. Jednocześnie artykuł ten jest zaproszeniem do dyskusji na temat "Konflikty AMIGI i PCta". Moim zdaniem nic tak nie podnosi poziomu adrenaliny w organiźmie Amigowca, jak nieodpowiedzialne wypowiedzi niektórych autorów, np.: "Nowa karta graficzna kosztująca jedyne x dolarów (gdzie x=> 4000$) zapewnia wspaniałe efekty video nieosiągalne na innych komputerach", świadczące najczęściej o kompletnym braku rozeznania w temacie.

Ponadto w numerze znajdziecie ankietę. Jest to szansa przekazanie nam opinii o kształcie pisma. Nie przegapcie tej okazji, być może zaowocuje to stałą współpracą z naszym wydawnictwem. Oczywiście najlepsze wypowiedzi nagrodzimy. Szczegóły na stronie 31.

Uległ zmianie numer kuponu konkursowego na naszej okładce. Od zeszłego numeru kupon jest opatrzony dwójką. Oto laureaci konkursu nr 1: Waldemar Kozłowski z Poniec, Janusz Walkiewicz z Chrzanowa, Marcin Cetera z Rączna, Igor Liszkowski z Wrocławia, Łukasz Grabaj z Dąbrowy Gór.

Zapraszam do lektury następnego numeru Amigowca.

R. Kowalski



9 **FASTLANE**
Kontroler SCSI II i rozszerzenie pamięci w jednym.



10 **KOPROCESOR DLA A1200**
Rozważania ne temat montażu koprocesora matematycznego do Amigi 1200. Pierwsza część artykułu o koprocesorze i jego zastosowaniu.



15 **PORÓWNANIE ODZYSKIWACZY**
Tomasz Hrycuniak porównuje trzy programy do odzyskiwania danych: Quarterback Tools, AmiBack Tools i Disk Salvage.



21 **OPUS V4.11**
Druga część artykułu Tomka Kokoszczyńskiego o niesamowitych możliwościach tego programu narzędziowego.

EDITORIAL

1 Od redakcji

2 Spis treści

DEPESZE

4 Depesze

70 Krótkie opisy programów i sprzętu

SPRZĘT

8 Amtari

9 Fastlane Z3

10 Koprocesor dla Amigi 1200

PROGRAMY

15 Porównanie odzyskiwaczy

21 Opus v4.11

MINI-MAX

25 Brilliance część 3

FELIETONY

28 Przeszłość...

48 Kwaśno o zasadach

TRYBUNA

34 Z życia...

ANKIETA

31 Ankieta

KURS

Kurs Asemblera 34

KONKURS

Rusz głową 39

PISMA, PISEMKA

Pisma, pisemka 40

PUBLIC DOMAIN

Amigowy Public Domain 42

SCENA

Wernisaż 00

Lista demonów 46

WARSZTATY

Dźwięk i muzyka w programach Amosa 51

Arexx część 3 54

Studio 57

Droga z gazetami 59

Imagine część 2 61

Supra fax-modem 14400 63

Samplowanie 65

Kwadratura koła 67

AMI-MARKET

Ami-Market 73

ANKIETA 31

Wypowiedz swoje opinie i uwagi na temat pisma. Mile widziane nowe pomysły i konstruktywna krytyka.



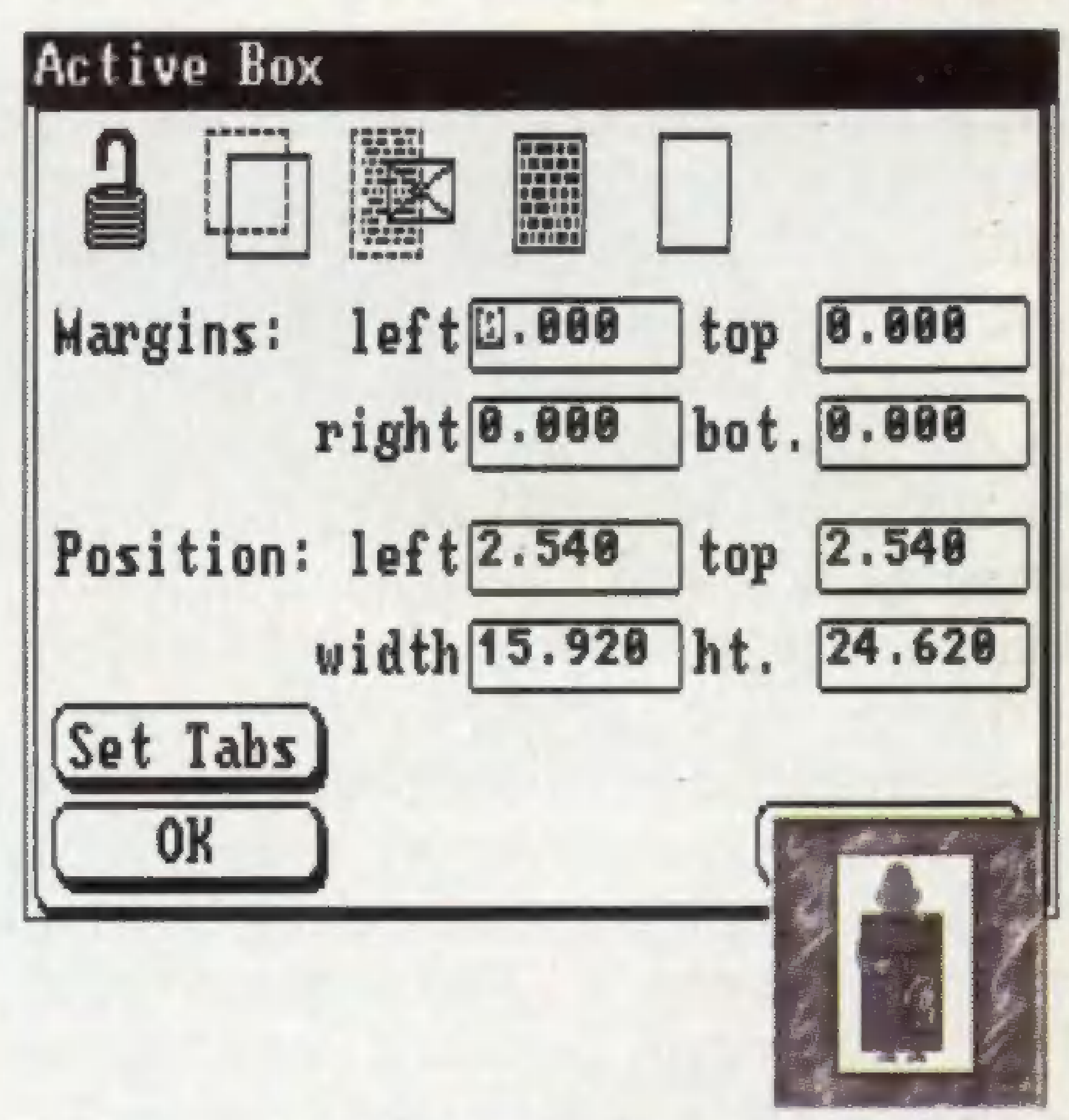
KWAŚNO O ZASADACH 48

Treścią artykułu są przykazania programisty. Jarek Chrostowski analizuje powstawanie błędów w technice programowania i podaje sposoby ich unikania.



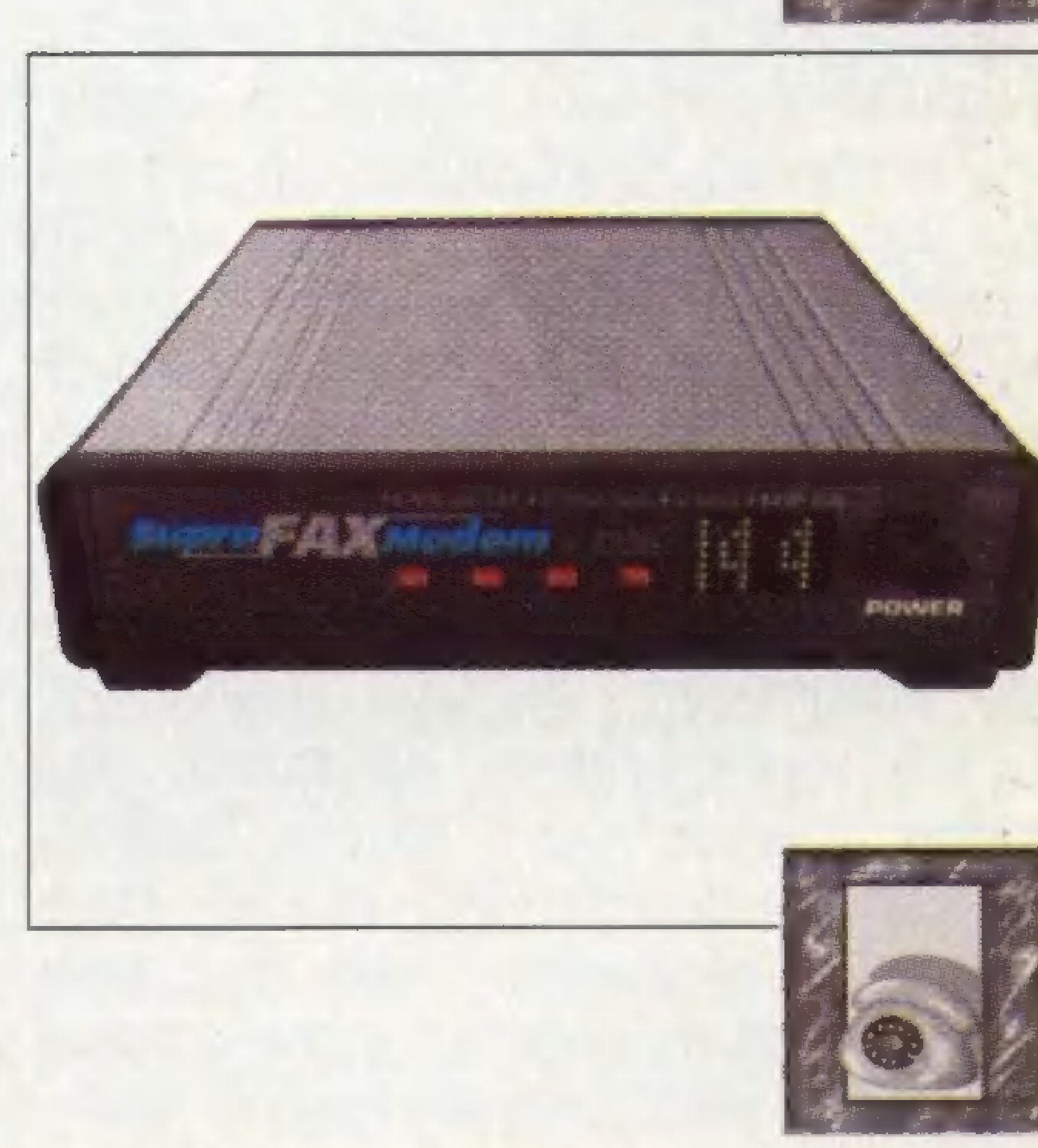
DROGA Z GAZETAMI 59

Tomasz Kokoszczyński przedstawia sposoby tworzenia najprostrzych gazetek na programie Page Setter.



FAX-MODEM SUPRA 63

Tomek Matajek przedstawia wspaniały fax-modem firmy Supra.



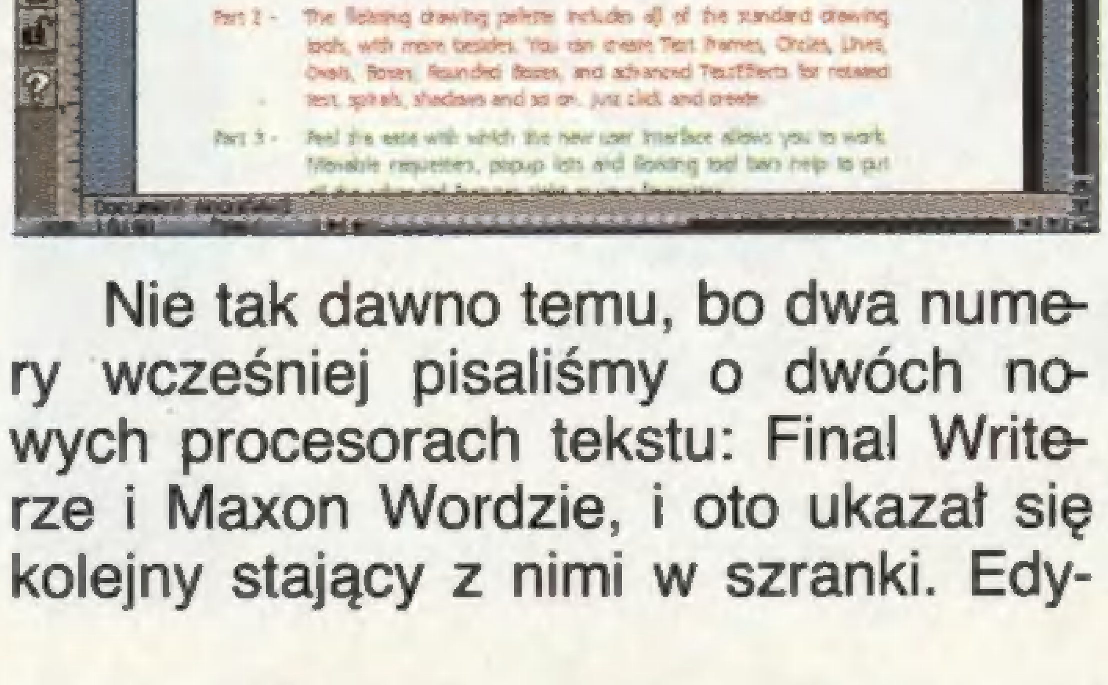
PageStream3



PageStream 3.0.

Od firmy Soft Logik otrzymaliśmy kolejną wersję alfa pakietu DTP PageStream 3.0. Po zapoznaniu się z programem można bez wahania stwierdzić, że jest to jeden z najlepszych programów DTP jakie dotychczas powstały. Posiada wszystkie zalety takich potentantów jak QuarkXPress 3.3 czy PageMaker 5.0 oraz wiele innych. Specjalna paleta ułatwiająca edycję zmienia swoje znaczenie w zależności od wybranej funkcji np. podczas edycji tekstu są na niej dostępne opcje formatowania takie, jak rodzaj i wielkość czcionki, kerning, tracking, odległość międzyliniowa, styl, justyfikacja i wiele innych. Program daje użytkownikowi możliwość wykorzystania dodatkowych palet pomocniczych m.in palety koloru, styli, czcionek, trapingu, makrodefinicji i zarządzania stronami. Wprowadzono tzw. Tool Bar, czyli pasek z najczęściej używanymi operacjami. PageStream 3.0 umożliwia tworzenie ramek z obrazkami o dowolnym kształcie, np. można umieścić zdjęcie w chmurce. Wypełnianie stworzonych obiektów możliwe jest na 6 sposobów: zwykłe (pełnym kolorem), wzorkiem, cieniowanie linearne, radialne, logarytmiczne oraz z dostosowaniem do kształtu. Kolory są obsługiwane w standardach CMYK, RGB, HSV, PANTONE i EfiColor, dzięki czemu ich reprezentacja ekranowa jest zgodna z wydrukiem. Można definiować traping dla każdego obiektu. Każda strona dokumentu może być tworzona na podstawie nieograniczonej liczby stron wzorcowych. Istnieje automatyczne kreowanie przypisów, wyszczególnionych punktów, dużych liter zaczynających rozdział, cudzysłowów otwierających i zamykających. Dodatkowo program umożliwia wczytywanie dokumentów stworzonych w pakietach Pro Page 4.1, QuarkXPress, Page Maker, Word Perfect i Final Writer. W modułach importu grafiki znajdą się takie pozycje jak Kodak PhotoCD, JPEG, Freehand EPS oraz co zadziwiające, będzie to pierwszy program DTP umożliwiający import plików Corel Draw (.cdr). Dodatkami do PageStream będą: edytor tekstu PageLinear 2.0, program do ob róbki grafiki bitowej BME 2.0, system komunikacji między programami HotLinks oraz 20 czcionek w formacie Adobe Type 1.

WordWorth 3.

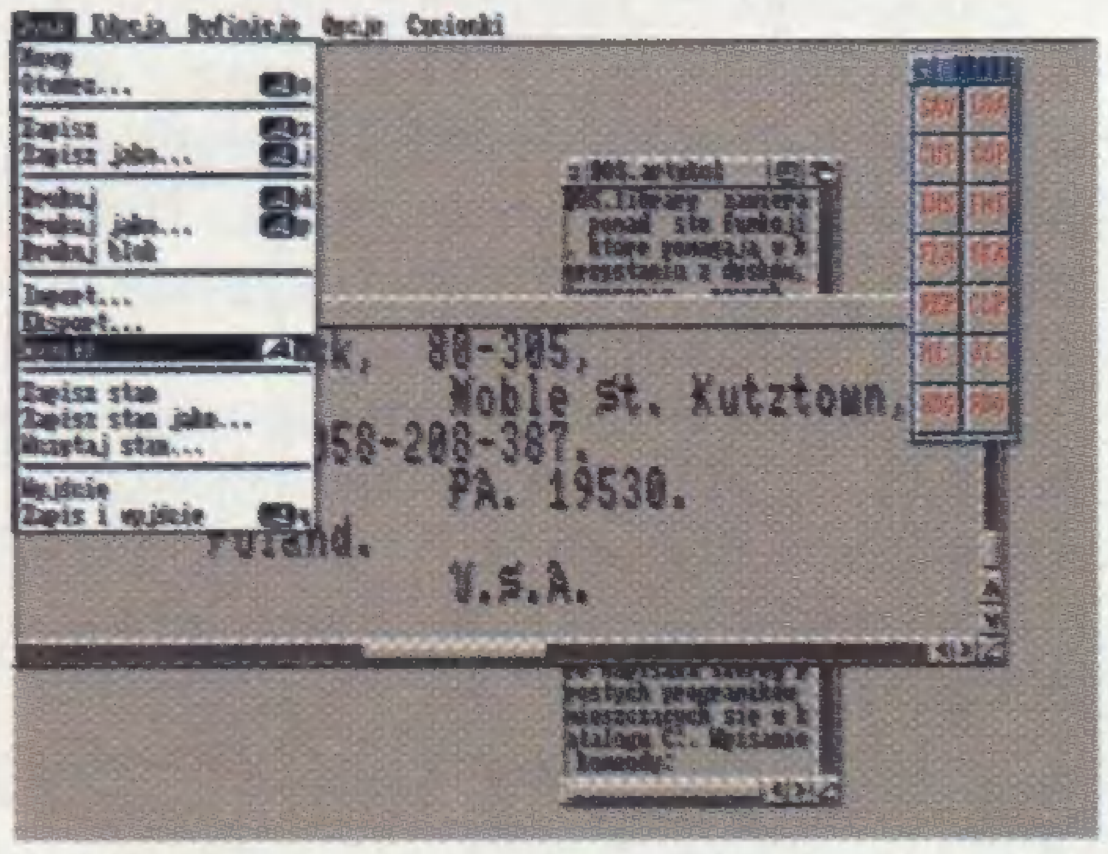


Nie tak dawno temu, bo dwa numery wcześniej pisaliśmy o dwóch nowych procesorach tekstu: Final Writerze i Maxon Wordzie, i oto ukazał się kolejny stający z nimi w szranki. Edytorem tym jest trzecia wersja znanego programu "Wordworth" angielskiej firmy Digita International. Jest to praktycznie napisany od nowa produkt. Posiada coraz częściej stosowany tzw. "Tool Bar", czyli pasek z najczęściej używanymi opcjami, którego pozycja, jak i wielkość jest dowolnie regulowana. Dodatkowo funkcje w nim zawarte, można ustawiać wedle własnych potrzeb. Wordworth rozpoznaje czcionki formatów Agfa® Compugraphic®, Adobe® PostScript™ i TrueType®. Specjalny moduł efektów umożliwia ułożenie napisów na łuku, okręgu, spirali, fali, itp. Oczywiście bezproblemowo można uzyskać elementy graficzne takie jak ramki, okręgi czy linie. Bardzo proste jest tworzenie spisu treści, indeksów oraz tabelek. Wystarczy tylko kliknąć na odpowiedniej ikonce i rozciągnąć ramkę, która określi jej wielkość. Program sam spyta się o ilość kolumn i wierszy oraz czy wstawić linie dzielące poszczególne pola. Nie zapomniano również o słowniku ortograficznym i synonimów, niestety oferowane są tylko wersje angielskie i niemieckie. Jednym słowem jest to kolejny świetny procesor tekstu na miarę Microsoft Worda czy AmiPro 3.0. Panowie pecetowcy, nie macie się czym chwalić!

Polacy nie gorsi...

Studio GEMINI dostarczyło nam zapowiedzi swoich nowych produktów. Pierwszy z nich to Edytor Tekstu. Jego pełna wersja będzie miała cechy wyznaczone przez równanie edytor = CED + NotePad + Okienka + Ansii. Będzie on przeznaczony dla użytkowników drukarek nielaserowych, programistów, wszystkich osób zajmujących się obróbką tekstu zapisanego w rozszerzonym formacie ANSII. Zgodnie z tymi założeniami będzie można łatwo zmieniać font, jak i jego wygląd (kursywa, podkreślenie itp.), jednak bez możliwości zmiany wielkości - edytor ma pracować w pełnym WYSIWYG, z tekstowym odwzorowaniem na drukarce. Będzie posiadał pewne ułatwienia dla programistów takie jak, automatyczne tworzenie wcięć w programie (dostarczone moduły będą potrafiły tworzyć wcięcia w C, Pascalu, Oberonie i E), automatyczna zmiana koloru i fontu określonych elementów tekstu, np. odróżnienie komend od komentarzy. Autorzy planują pełną konfigurowalność edytora, od menu i toolboxa na komunikatach kończąc. W ten sposób dołączenie rozkazu "kompiluj" nie będzie problemem. Program ma posiadać istotne udoskonalenia względem CEDa. Przede wszystkim będzie korzystał ze standardowe

go systemu okienek oraz ... będzie jeszcze szybszy niż CED!!! Dużym usprawnieniem będzie korzystanie z buforów dyskowych (opcja dla posiadaczy HD) do przechowywania dokumentów o dowolnych rozmiarach maksymalnie do 4 GB. Nieskończony poziom Undo + Redo (również bufory) pozwoli na bezpieczne tworzenie dowolnych dokumentów. W planach są proste tabelki - chociaż jest to sprzeczne z założeniami początkowymi (ANSII). Poniżej prezentujemy ekran programu. Jest to oczywiście obraz początkowego stanu prac i efekt końcowy będzie na pewno inny. Z całą pewnością zmieni się układ gadżetów oraz ich funkcje, to samo dotyczy menu. Toolbox zostanie wypełniony przez odpowiednie ikonki. Planowane wydanie kwiecień 1994.



Kolejny planowany produkt to program DTP Skład!!! Ma być to zaawansowany pakiet pracujący w sieci. W skład zestawu wejdzie program główny, program do obsługi sieci oraz sama sieć. Bliższe szczegóły wkrótce. Planowane wydanie wrzesień 1994.

W testach znajdują się jeszcze dwa programy. Budujemy Teleskop Zwierciadlany to pierwszy z przewidywanej dłuższej serii programów z cyklu Biblioteka Miłośnika Astronomii. Program zawiera pełny opis, wraz z ponad 40 rysunkami i animacjami, przedstawiający wszystko, co jest niezbędne przy budowie amatorskiego teleskopu. Akord v1.2 to program do nauki gry akordowej na gitarze. Zawiera 14 lekcji, począwszy od dokładnego zapoznania się z instrumentem, na egzaminowaniu użytkownika na podstawie zagranej melodii skończywszy. Oba programy są przewidywane na marzec 1994.

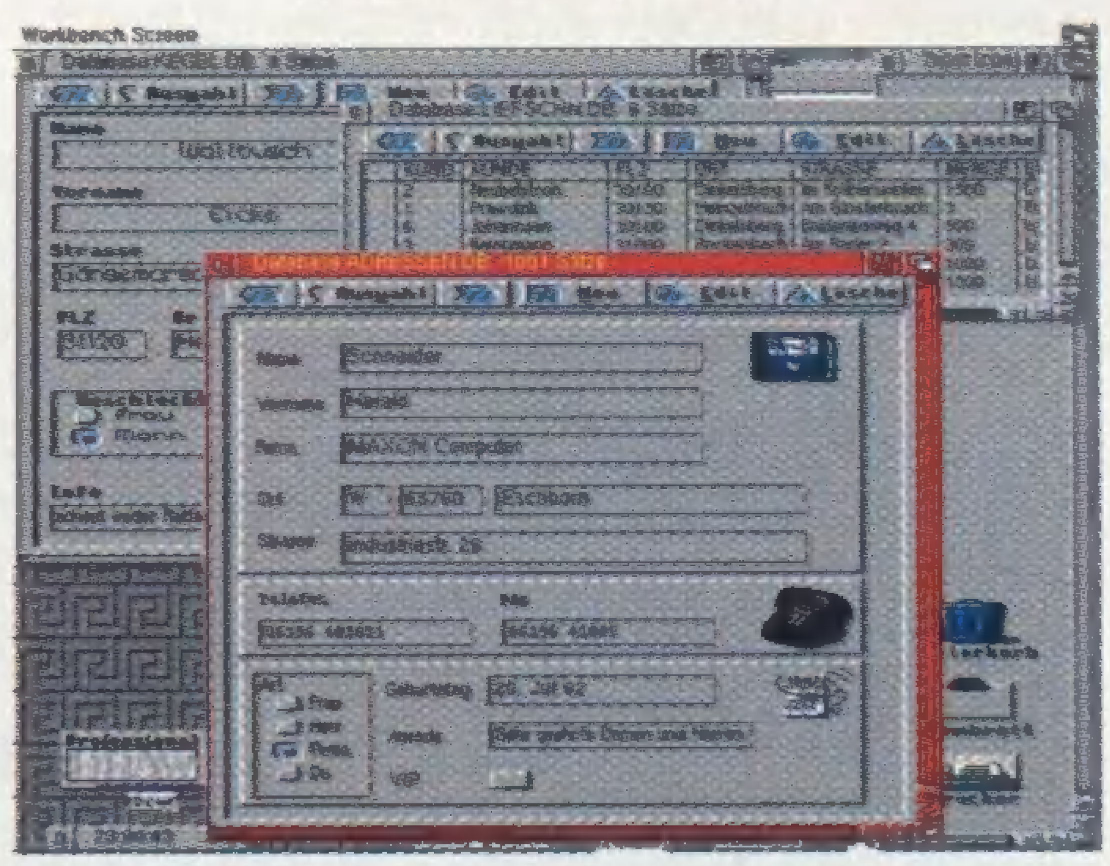
Teraz coś o sprzęcie. 12-bitowy sampler - nowy produkt GEMINI ma być podłączany do złącza drukarki. Początkowo nie będzie niestety przelotowy. Przewidywane są wersje stereo i mono. Parametry: 12 bitów na kanał, częstotliwość próbkowania do 44 kHz, korekcja własnym filtrem dolnoprzepustowym. Do samplera będzie dołączany prosty program samplujący oraz prawdopodobnie (obecnie prowadzone są rozmowy) najnowsza wersja ProTrackera z 4 kanałami dźwięku 12-bitowego. Data wejścia na rynek kwiecień 1994.

Firma zapowiada rewelację. Pierwszą w Polsce kartę turbo z procesorem Motorola 68060. Obecnie trwają testy i próby. Pierwsze egzemplarze mają być osiągalne około maja 1994.

Jak widać plany ambitne. Zobaczymy czy starczy sił do ich realizacji. Dla zainteresowanych podajemy adres firmy:

STUDIO Gemini,
ul. Wita Stwosza 73 p. 111,
80-305 Gdańsk,
tel. 52-13-12 lub 52-45-11 wew. 25,
fax 20-83-87

Twist dla każdego.



Firma Maxon wypuściła na rynek swój następny produkt. Tym razem jest to "prosta" baza danych. Prosta jeśli chodzi oczywiście o obsługę. Pewne pojęcie o bazach danych należy oczywiście mieć. Choćby takie, co to takiego i do czego się ich używa. Opanowanie samego programu nie powinno przysporzyć nikomu faktycznych trudności. Wiele operacji wykonuje się jak w zwykłym magnetofonie. MaxonTWIST nie jest jednak taki prosty jeśli chodzi o możliwości. Jest to szybka baza o dużym obszarze zastosowań zarówno w domu, jak i w pracy. Pola mogą być łączone ze sobą na wiele różnych sposobów. Bardzo obiecujące są funkcje wyszukiwania danych, a to przecież w bazach danych rzecz najważniejsza. Można stosować wiele filtrów, szukać fragmentów tekstu, itd. Tylko przy pomocy myszki można sortować, tworzyć dowolne zestawy danych, etykietki, czy listy wielokrotnego użytku (ta sama treść, a zmienny adres). Program oczywiście współdziała z ARexxem, tak że przenoszenie danych wszelkiego typu i zarządzanie a'la karte nie stanowią żadnego problemu.

Nowy Quarterback.

Quarterback jest programem używanym przez większość posiadaczy twardych dysków. Twardy dysk jest wspaniałym nośnikiem, ale ma jedną zasadniczą wadę - jeśli już "padnie", to można stracić całą jego zawartość. Czasami może się na nim znajdować nasza wieloletnia praca i jeśli się okaże, że akurat przez cały ten czas nie mieliśmy nastroju do robienia sobie kopii co ciekawszych programów, to może być naprawdę kiepsko. Otóż Quarterback jest programem służącym do archiwizowania twardego dysku. Można przy jego pomocy przepisać zawartość twardego dysku. Zarówno całego, jak i poszczególnych, wybranych plików. Można te pliki nawet lekko podpakować. Nie polecam jednak nikomu tej opcji, ponieważ program ma potem kłopoty ze znalezieniem pliku, który nas interesuje i przegląda po kolei wszystkie dyskietki. Krótko mówiąc - archiwizowanie na dyskietkach, jest nie tylko nudne, ale również mało wydajne. Wyobraźmy sobie przepisanie dysku 200MB na 200 dyskietek. Jeżeli uda nam się to w ciągu jednego dnia, to i tak sukces. Nowa wersja programu 6.0 pozwala na magazynowanie danych na innych, dowolnych nośnikach - np. na taśmie magnetycznej w tzw. streamerach. Jest to niezwykłe udogodnienie szczególnie dla dużych firm. Inną możliwością jest stworzenie odpowiednich skryptów w ARexxie i magazynowanie danych pod ich kontrolą - tak, że zostaną zmagazynowane tylko te dane z tych katalogów, o które nam chodzi. Tym sposobem nie musimy za każdym razem martwić się, że coś pominęliśmy. W dobie przechodzenia Amigi na pełen profesjonalizm program taki jest faktycznie nieodzowny.

Amiga idzie na Video.



Ostatnio wszystkie pisma komputerowe poświęcone naszej przyjaciółce zachłystują się jej możliwościami graficznymi. Nie od razu Kraków zbudowano, tak samo nie od razu nasz ulubiony komputer wyrobi sobie markę w tej dziedzinie. Tematowi amigowej

grafiki poświęcony został specjaln, zorganizowany przez Commodore i RTL show, który odbył się 13 stycznia 1994 w jednym ze studiów filmowych znajdujących się w Hamburgu. Show był zatytułowany "seaQuest DSV" i była to podróż do roku 2018. Pojazdem, który przepływał przez głębiny była właśnie łódź podwodna "seaQuest DSV". Okazją do zorganizowania pokazu jest serial stworzony przez Stevena Spielberga. "SeaQuest DSV" jest obecnie nadawany w programie RTL. Oczywiście znaczna część filmu (ok. 40%) to zdjęcia trikowe wykonane na Amidze. Film był tworzony na 70 Amigach 4000/040 przez 35 osób. Wypada średnio 2 Amigi na łebka. I słusznie. Na jednej się pracuje, druga zaś liczy.

Na marginesie dodam, że niektórzy posiadacze zestawu Silicon Graphic z programem SoftImage oferowanym w Polsce przez firmę ATM za jedyne 280 tysięcy dolarów nie są w stanie zrozumieć, że prawie identyczne efekty jak na ich sprzęcie można osiagnąć przy użyciu zwykłych Amig. Gdybym wydał tyle pieniędzy, też nie byłbym w stanie.

Serdeczne pozdrowienia dla pracowników pracowni grafiki komputerowej UW, którzy na targach Komputer '94 tak dzielnie kłócili się z jednym z naszych redaktorów, nie mając pojęcia o czym mówią!

Kompakty raz jeszcze.

Dzięki Amidze CD32 tworzy się prawdziwa moda na Amigowe kompakty. Firmy, jak to już uprzednio wspominaliśmy, wypuszczają coraz więcej dysków, które zawierają wszystko co tylko da się wymyśleć. Niesamowita pojemność powoduje, że na jednym takim dysku można upchać dzieła zebrane Szekspira, Marksa i kilku pomniejszych autorów. Fascynująca jest możliwość dostępu do takich danych - ciach, mach i gotowe. Pamiętajcie jeszcze, że samo posiadanie nawet najbogatszego zbioru dysków kompaktowych na Amigę jeszcze o niczym nie świadczy. Czasami nie warto być głupszym od własnego komputera.

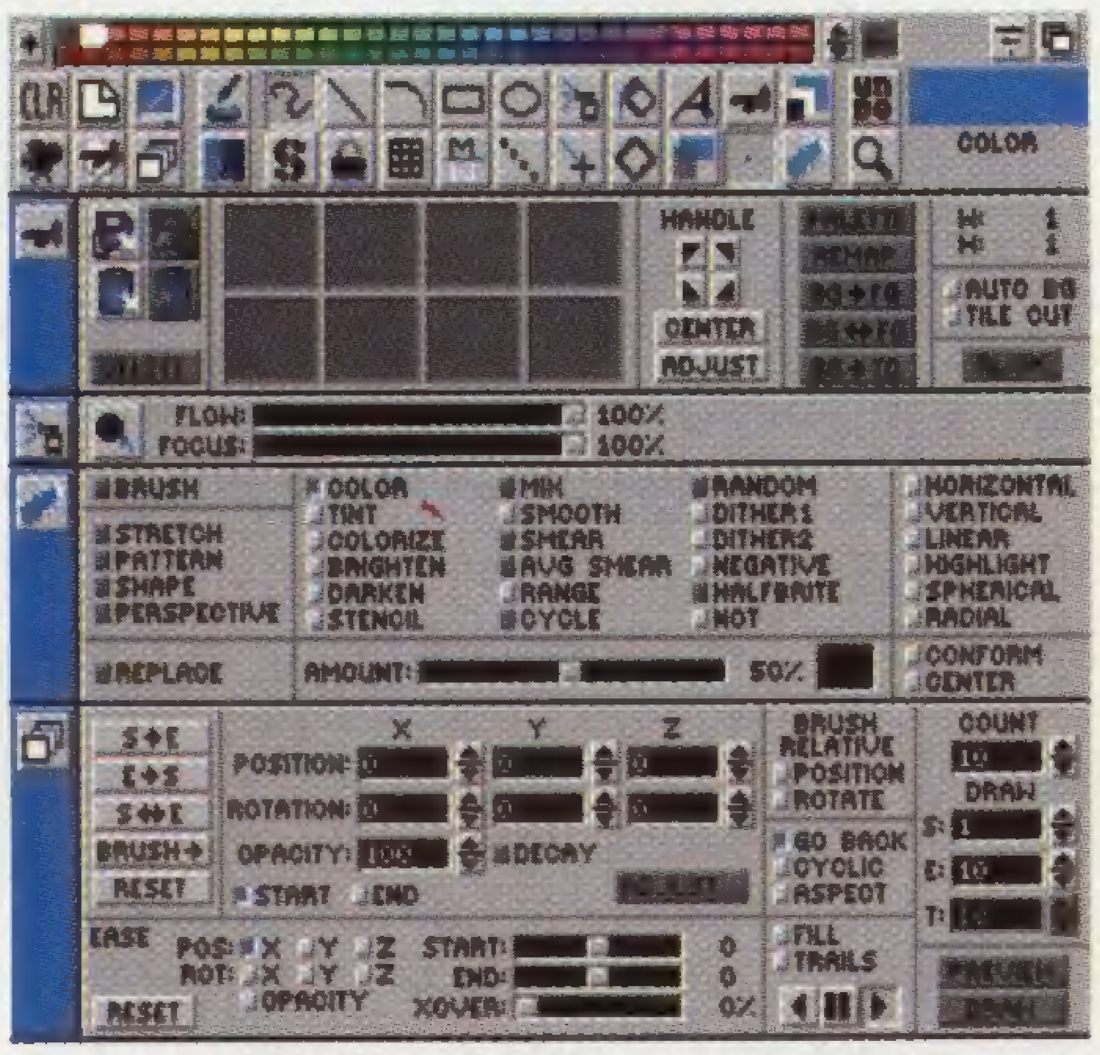
Video w ruchu, czyli FMV

FMV to skrót od nazwy Full Motion Video. Jest to dostawka do Amigi CD32, która powoduje, że Amiga ta staje się nie tylko domową biblioteką, zabawiaczem dzieci i na dodatek zwykłym komputerem, ale pozwala na odtwarzanie filmów video z płyt kompaktowych, co do tej pory nie było możliwe. Oczywiście jakość obrazu z takiego urządzenia jest nieporównywalna do



jakości najlepiej nawet nagranych filmów video. Jako demonstracja możliwości nowej przystawki, do sprzedawców (raczej tych zagranicznych) zostanie rozesłany dysk z koncertem zespołu Eurytmics, "Rise of the Robots" Mirage'a i "Inferno" D.I.D.a.

Brilliance - opinie



W grudniowym numerze AMIGOWCA opisaliśmy program Brilliance. Jest to program "artystyczny" w pełni współpracujący z nowymi układami graficznymi AGA. Na pierwszy rzut oka wydaje się on nieco dziwny. Dziwna jest obsługa, dziwne (w znaczeniu pozytywnym) są jego możliwości. Jednak z doniesień prasowych poświęconych temu programowi wynika, że zaczyna on powoli zastępować DPaintaIV. Nikt nie twierdzi, że wyprze go całkowicie. Prostsze grafiki ciągle będzie łatwiej tworzyło się w DPIV. Jednak Brilliance oferuje nowe jakości i wszystkich początkujących oraz bardziej zaawansowanych grafików namawiamy do wypróbowania programu. Specjaliści twierdzą, że warto. Pojawiła się już wersja 2.0 tego programu. Szczegóły wkrótce.

Nowa dopalarka do 1200-tki!

Firma GVP słynie z wielu dobrych produktów dla naszych przyjaciółek. Firma powstała w roku 1988, a w tej chwili jest największym dostawcą peryferii do Amigi. W jej ofercie królują (oczywiście nie tylko) twarde dyski oraz karty turbo. Obecnie firma wprowadziła na rynek jedną z najlepszych, jeśli nie najlepszą kartę trubo do Amigi 1200. Karta nosi nazwę "A1230 Turbo+ Performance Series II" i została wyposażona w procesor 1230 taktowany zegarem 50 MHz. Dostępna jest także wersja taktowana zegarem 40 MHz. Karta posiada oczywiście miejsce na koprocesor i dodatkowe 32 MB pamięci RAM, co umożliwiłoby zrobienie z naszej małej przyjaciółki prawdziwie profesjonalnego smoka. Prawdziwą innowacją jest jednak port DPP (DMA Peripherial Port), który umożliwia użytkownikom tej karty korzystanie z SCSI II, digitizerów czasu rzeczywistego, 16-bitowych digitizerów dźwięku oraz z wielu urządzeń wymagających szybkiego przesyłania danych do Amigi. Pierwszym urządzeniem wykorzystującym ten port, a dostępnym w chwili obecnej na rynku jest GVP A1291 Kit, czyli dodatkowy port SCSI II umożliwiający przesyłanie danych z szybkością 4,2 MB/sekundę! Karta jest bardzo łatwa w montażu i obsłudze, a przy okazji wcale nie narusza pieczęci gwarancyjnych.

Dyski wędrowniczki

Szczególnie w profesjonalnych zastosowaniach Amigi musimy borykać się z trudnością przenoszenia dużych ilości danych z jednego komputera do innego. W redakcji dajemy sobie radę za pomocą coraz popularniejszych w naszym kraju SyQuestów, czyli twardych dysków z wymiennym nośnikiem. Ostatnio amerykańska firma SyQuest Technology Ltd. zaprezentowała kolejny produkt - stację dysków magnetycznych o pojemności 270 MB. SQ 3270 jest kompatybilna z będącym od niedawna w sprzedaży napędem SQ 3105 (105 MB). Osiągi nowego SyQuesta są porównywalne z aktualnie produkowanymi dyskami twardymi (czas dostępu poniżej 13 ms.) Nowe media charakteryzują się lepszą odpornością na wstrząsy, niż to było w dotychczasowych wersjach. SQ 3270 powinien się pojawić na rynku wiosną 1994.

Tani dopalacz dla szybkich?

Włoska firma Hardital wypuściła na rynek dopalacz do Amigi 3000 i 4000/030 o konstrukcji opartej na procesorze MC 68040. Komputer wyposażony w taką kartę powinien pracować ok. 20% szybciej niż A4000/040. Cena karty nie jest nam jeszcze znana, ale

krążą plotki, że będzie ona wyraźnie niższa, niż inne produkty tej klasy.

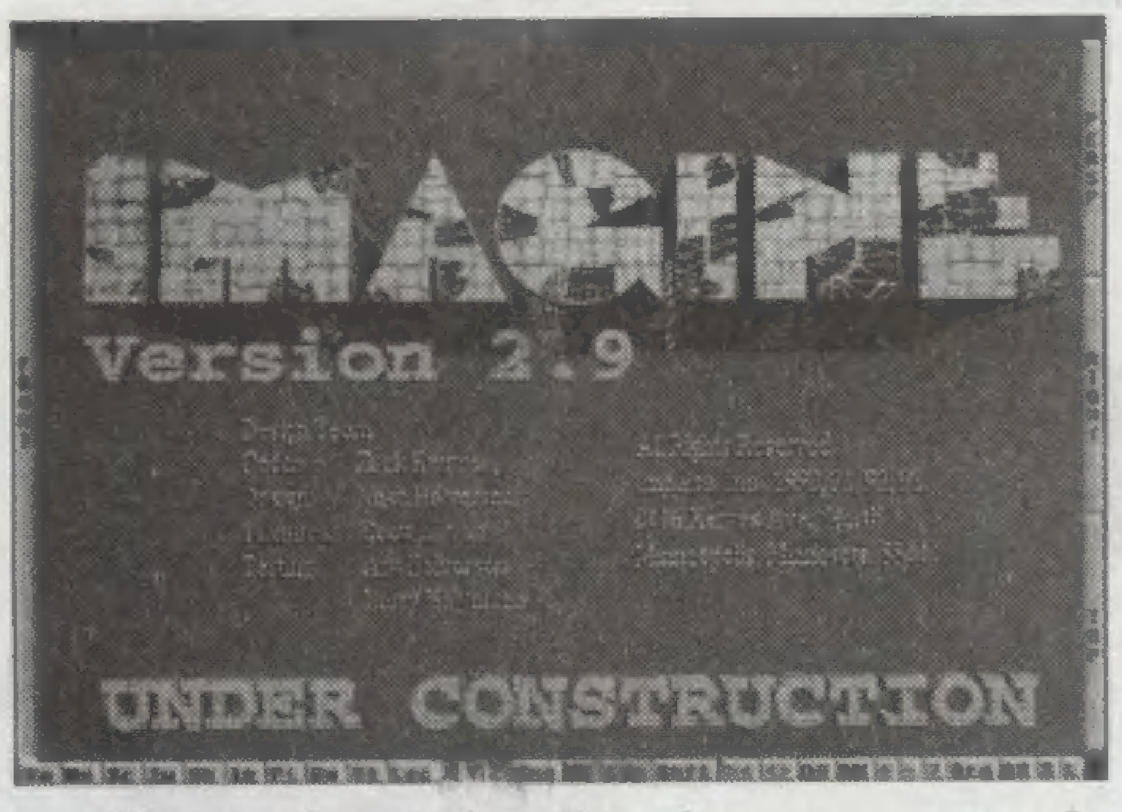
Znowu GVP.

GForce 040/40, to oznaczenie najnowszego dopalacza do Amigi 4000 firmy GVP, Posiada on na płycie procesor MC 68040 taktowany zegarem 40 MHz i kontroler SCSI II, pozwalający na osiągnięcie prędkości transmisji rzędu 10 MB/s (oczywiście z odpowiednim dyskiem). Podstawka, w której umieszczony jest procesor, umożliwi w przyszłości wymianę procesora na MC 68060. Karta posiada miejsce na 128 MB RAM w postaci modułów GVP SIMM lub standardowych pamięci DRAM.

Dopalenie dla wolnych...

Wielu użytkowników Amigi 2000 przeklina obecnie chwilę, w której nabyli ten komputer. Ani to to do zabawy, ani do pracy. Owszem, jego szybkość nie jest wystarczająca, ale można go przecież wyposażyć w kartę turbo. Jedną z propozycji prezentowaną przez ARAM jest karta G-Force '030 Combo. Karta ta nie tylko przyspiesza komputer do szybkości osiąganej przez procesory 030, ale umożliwia wyposażenie go w 64 MB 32-bitowej pamięci, a dzięki interfejsowi SCSI II możemy do naszej przyjaciółki podłączyć do siedmiu urządzeń typu: SyQuest, CD-ROM, skaner, drukarka wyposażona w taki port, itp.

Imagine 2.9 - zapowiedź "trójki"?



Na rynku pojawiła się nowa wersja programu Imagine oznaczona jako 2.9. Wersja ta ma wiele nowych cech, które zostaną wykorzystane w wersji 3.0 i służy właściwie jako jej tester. W chwili, gdy do Państwa docierają te słowa wersja 3.0 powinna już być na rynku. Najważniejszą cechą jest nowy interfejs działający w czasie rzeczywistym. Umożliwia to bardzo wygodne i przystępne ustawianie elementów na scenie animacji. Wprowadzono niesamowite ułatwienie dla tych, którzy mają dostęp do figur generowanych przez AutoCADa - najpopularniejszy program tego typu na IBMa. Istnieje możliwość bezpośredniego wczytywania takich figur, a więc można teraz korzystać z niesamowitej biblioteki jego obiektów.

Wprowadzono wiele nowych sposobów deformowania obiektów, których brak dał się mocno odczuć w wersji 2.0. Obiekty można teraz wyginać, rozciągać, naciągać, przekręcać i demolować na wiele innych sposobów, zarówno przy użyciu myszy, jak i specjalnych opcji. Obiekty można teraz tworzyć z mniejszych figurek - np. z bąbelków. Bardzo łatwo tworzy się eksplozje. Nowe są funkcje definiowania obiektów, nowe są także funkcje ich grupowania. Możliwe jest tworzenie stanów jednego obiektu - np. człowieka w ruchu. Nie musimy teraz tworzyć kilku obiektów różniących się drobnymi szczegółami, ale tworzymy jeden obiekt, który zawiera w sobie te wszystkie stany, rewolucyjnie ułatwiono tworzenie napisów. Za pomocą spline editora możemy po prostu załadować dowolną czcionkę postscriptową i wpisać dowolny napis - oczywiście wszystko odbędzie się w trzech wymiarach. Zmieniło się także zarządzanie systemem ruchu obiektów. Nie musimy rysować mozolnych ścieżek dla każdego z nich - możemy je po prostu pogrupować. Dużym ułatwieniem jest możliwość sprzężenia kamery z dowolnym obiektem. Daje to naprawdę niesamowite efekty wizualne! Innym uproszczeniem życia licznych animatorów jest zastosowanie techniki warstwowej. Kilka obiektów można załadować do jednej warstwy, inne do drugiej. Aby nam się nie zrobił bałagan możemy sobie teraz jedną z warstw po prostu wyłączyć i zająć się spokojnie przestawianiem "pudełek" w drugiej.

Dodatkowymi cechami nowego Imagine, z których zapewno ucieszy się wielu grafików, jest możliwość pracy w trybie HAM8 oraz współpraca z kartą OpalVision. Jedną z niedogodności jest brak poprawy szybkości programu, no ale nie można mieć wszystkiego na raz. ❑

Magazyn Amiga (edycja niemiecka) opublikowała wyniki przeprowadzonej wśród swoich czytelników ankiety na najlepszy osprzęt i oprogramowanie do Amigi roku 1993. Jej wyniki przytaczamy poniżej:

Kategoria	Zwycięzca	Producent
Hardware		
Kontroler HD	Fastlane Z3	Advanced Systems & Software
Rozszerzenie RAM	Blizzard 1200	Advanced Systems & Software
Karta 68030	G-Force 030	GVP
Karta 68040	G-Force 040	GVP
Emulator	Emplant	Blue Moon
Flicker Fixer	DeInterlace Card	MacroSystem
Modem	ZyXEL U1496 EG+	ZyXEL
Digitizer	Vlab	MacroSystem
Genlock	DigiGen II	PBC Piet
Drukarka igłowa	Fujitsu DL 1100C	Fujitsu
Drukarka atramentowa	HP DeskJet 550C	Hewlett Packard
Drukarka laserowa	HP LaserJet 4L	Hewlett Packard
Rozszerzenie do A1200	A1230 Turbo	GVP
Rozszerzenie do A4000	Fastlane Z3	Advanced Systems & Software
Software		
Edytor tekstów	Final Copy II	Soft Wood
Baza danych	Superbase 4.0	Oxxi
Program muzyczny	Bars & Pipes	Octave 2
DTP	Page Stream	Soft Logic
Program malarski	DPaint IV AGA	Electronics Arts
Program animacyjny	ClariSSA 2.0	pro DAD
Ray-Tracing	Real 3D	Activa International
Program prezentacyjny	Scala MM	SCALA luc.
Program narzędziowy	(Tool) Directory Opus	Inovatronics

AMTARI

Tomasz Kulbacki

Kolejna wersja znanego już emulatora Atari ST/TT odróżnia się od swoich poprzedników głównie specyficznymi wymaganiami sprzętowymi, eliminującymi 90% potencjalnych użytkowników. W przeciwieństwie do poprzednich wersji, najnowsza odmiana tego popularnego emulatora nie jest rozwiązaniem czysto programowym, lecz kombinacją sprzętowo-programową.

Amtari w swej najnowszej odmianie wymaga min. Kickstartu 2.04, procesora od MC68020 w górę, MMU i slotu ZORRO II lub III. Jedynymi komputerami, które spełniają powyższe wymagania są: Amiga 2000 z kartą turbo (nie z każdą) i Amiga 3000. A4000/030 wypada z gry z powodu posiadania procesora pozbawionego MMU, natomiast A4000/040 nie może funkcjonować z Amtari ze względu na to, że Atari, jak dotychczas, nie zdołało opracować wersji systemu działającej z procesorem 040.

Amtari 1.70 jest sprzedawane w pakiecie składającym się z dyskietki z oprogramowaniem, karty ZORRO II z podstawkami pod ROMy z TOS'em i 22-stronicową instrukcją obsługi (po niemiecku). Teoretycznie, emulator mógłby działać bez tej karty. Potrzebny byłby w tym celu programik umożliwiający przepisanie TOSu z ROMu na dyskietkę. Instrukcja informuje, że taki program został dołączony, jednak moje poszukiwania na dyskietce nie zostały uwieńczone sukcesem. Pozwala to przypuszczać, że Producent emulatora zdecydował się na kombinowane rozwiązanie programowo-sprzętowe pod naciskiem Atari, które nie chciało, aby istniała potencjalna możliwość użytkowania TOSu bez jego zakupu w oryginalnej wersji.

Instalacja emulatora jest nieskomplikowana i nie wymaga żadnej specjalnej umiejętności. Po zamontowaniu karty w slocie, należy odpalić znajdujący się na dyskietce program instalujący (Hardware nie konfiguruje się automatycznie w systemie) software'ową część emulatora na twardym dysku, bądź też wystartować emulację bezpośrednio z dyskietki.

Amtari współpracuje bezproblemowo z twardym dyskiem Amigi. W przeciwieństwie do np. Chamaeleona, nie wymaga żadnych driverów, gdyż partycja emulatora jest, od strony AmigaDOS'u, zwykłym plikiem. To bardzo uniwersalne rozwiązanie powoduje jednak spore straty prędkości podczas pracy.

Ponieważ TOS potrafi zarządzać tylko dwoma stacjami dysków, Amtari także posiada to ograniczenie. W preferencjach emulatora istnieje możliwość zdefiniowania, które z ze stacji DF0: - DF3:, mają emulować atarowskie napędy A: i B:. Stacje dysków HD funkcjonują bez zarzutu.

Amtari wykorzystuje wszystkie tryby graficzne ECS. Emuluje bez zarzutu grafikę Atari ST we wszystkich rozdzielczościach. Dodatkowe tryby, które są dostępne w Atari STE i TT nie funkcjonują z Amtari. Multitasking, w przeciwieństwie do starszych wersji emulatora, nie podlega żadnym ograniczeniom.

Emulator ma dostęp do złącza szeregowego i równoległego Amigi. Teoretycznie może sterować także interface MIDI poprzez port RS-232, lecz w praktyce bywa z tym bardzo różnie.

Amtari charakteryzuje się stosunkowo dobrą kompatybilnością w porównaniu z konkurencyjnymi produktami, choć jego oprogramowanie pozostawia jeszcze co nieco do życzenia. Interesujący jest fakt, że programy funkcjonują na nim szybciej, niż na oryginalnym Atari (przy takim samym procesorze), co jest z pewnością jego ogromną zaletą.

Amtari 1.7 można nabyć w RFN za ok. 400 DM (z oryginalnym TOS'em). ❑

FASTLANE Z3

Tomasz Kulbacki

Odejście Commodore od lansowanego w modelu A3000 standardu SCSI spowodowało falę krytyki pod adresem firmy, ale jednocześnie otworzyło szerokie pole dla działalności firm, pragnących skorzystać z faktu, że nie wszyscy właściciele Amig 4000 zadowolą się korzystaniem z wbudowanego kontrolera AT-BUS.

Pierwszą z firm, która zdecydowała się zapełnić powstałą na rynku lukę i wypuścić kontroler SCSI przystający swoimi parametrami do Amigi 4000 była niemiecka firma Phase 5 Digital Products sprzedająca swoje produkty poprzez znanego na rynku niemieckim i amerykańskim dystrybutora: Advanced Systems & Software.

Fastlane Z3 - bo o nim tu mowa - jest kontrolerem FAST-SCSI II pozwalającym na transfer danych z prędkością do 10 MB/s (synchronicznie) I jednocześnie kartą rozszerzającą pamięć RAM do 64 MB. Po wymianie dwóch układów jest możliwe nawet uzyskanie 256 MB na płycie kontrolera. Jako ciekawostkę podam fakt, że przy obecnych cenach pamięci w pełni obsadzona karta jest warta mniej więcej tyle, co średniej klasy samochód osobowy. Na karcie można instalować standardowe pecetowskie ośmio lub dziewięciobitowe moduły SIMM o pojemności 1 lub 4 MB. Po zamianie uprzednio wspomnianych układów można instalować już tylko SIMMy 16 MB.

Kontroler jest sprzedawany w bardzo estetycznym opakowaniu zawierającym dodatkowo dyskietkę z oprogramowaniem instalacyjnym, kabel SCSI umożliwiający podłączenie dwóch urządzeń wewnętrznych, kość o symbolu 74FCT240 (o tym później) i starannie wydaną, bardzo przejrzyście napisaną (dla znających niemiecki) instrukcję instalacji i obsługi sprzętu oraz dołączonego oprogramowania. Od pewnego czasu firma dodaje gratis program DynamiCache znacznie przyspieszający pracę z twardymi dyskami.

Fastlane Z3 przeznaczony jest zasadniczo do pracy z Amigą 4000 i szybkimi dyskami twardymi SCSI II. Jest on oczywiście kompatybilny "w dół" i praca z normalnymi urządzeniami SCSI (SyQuest, CD-ROM, napędy magnetooptyczne, streamery) nie sprawia mu żadnych kłopotów, oczywiście w tym przypadku prędkość 10 MB/s pozostaje tylko na papierze. Zachowana jest kompatybilność ze standardem RDB (Rigid Disk Block).

Pewne niekonsekwencje popełnione przez Commodore powodują, że mogą wystąpić problemy w użytkowaniu kontrolera z pewnymi wersjami Amigi. I tak, praca z A3000 (4000 też) jest w ogóle niemożliwa, jeśli nie posiadacie na płycie kości Super Buster wersji -09 lub większej. Do pełnego wykorzystania możliwości Fastlane niezbędny jest Buster wersji -11. Nie przyspiesza on co prawda transmisji danych, lecz jego zastosowanie pozwala na efektywniejsze wykorzystanie czasu procesora.

Inny problem występuje w Amidze 4000 Rev. 3.0. Jeśli posiadacie komputer z taką płytą, będziecie musieli wymienić kość o symbolu 74FCT240 (dołączona).

Do Fastlane Z3 dołączone jest stosunkowo bogate oprogramowanie. Programem podstawowym, spełniającym taką samą rolę jak commodorowski HDToolbox, jest SCSIConfig. Nawiasem mówiąc przy pomocy HDToolbox można równie skutecznie wykonać większość czynności (np. rozpoznawanie, formatowanie i partycjonowanie dysków), do których producent przeznaczył SCSIConfig. Liczne programiki (CLI Tools) znajdujące się w katalogu "bin" pozwalają na efektywniejszą pracę kontrolera. I tak np.: program PatchATDevice przerabia system operacyjny (Kickstart 3.0 tak, że kontroler AT-BUS przestaje być rozpoznawany przez system (działa to tylko wtedy, gdy system jest ładowany z dysku), z kolei MSDOSMount umożliwia pracę z dyskami HD sformatowanymi pod MS-DOS, co ma kolosalne znaczenie przy transferze danych.

Podsumowując Fastlane Z3 jest bardzo przydatnym uzupełnieniem Amigi 4000. Dopóki nie pojawi się na rynku Amiga 4000 Tower, zakup Fastlane będzie jedynym rozwiązaniem dla posiadacza A4000 chcącego wyposażyć swój komputer w dużą ilość pamięci RAM i szybki twardy dysk. Cena takiego rozwiązania (ok. 900 DM) klasyfikuje jednak Fastlane jako sprzęt dla profesjonalistów. ❑



KOPROCESOR dla A1200

Wojciech Czyż

W jednym z poprzednich numerów Amigowca opisywałem sposób podłączania dysku twardego do A1200 - miało to podnieść komfort pracy. Jeśli uważasz, że osiągnięta w ten sposób prędkość komputera Cię nie zadowala, spróbuj nieco przyspieszyć jego operacje.

Na prędkość każdego komputera wpływ mają trzy czynniki:
- prędkość procesora głównego,
- ilość dostępnej pamięci,
- obecność specjalizowanych układów, które odciążają procesor główny.

Amiga od zawsze posiada zestaw specjalizowanych koprocesorów, które pomagają w operacjach graficznych oraz przy odtwarzaniu muzyki. W Amidze 1200 mamy możliwość montażu koprocesora matematycznego. Commodore przypadkowo ustaliło dość praktyczną zasadę - kupujesz A1200 z fabrycznie montowanym koprocesorem lub dokładasz go później na dodatkowej karcie wkładanej "pod klapkę". Tak więc jesteś zmuszony do kupna dodatkowej karty, a te wcale nie należą do tanich (przynajmiej chwilowo).

Fabryczne miejsce na koprocesor pozostaje niezagospodarowane, co daje możliwość jego wykorzystania (bez kupna dodatkowych kart). Niestety, operacja montażu koprocesora wymaga otwarcia obudowy i utraty gwarancji. Wchodzi jeszcze w rachubę nakłonienie autoryzowanego serwisu do jej wykonania. Zanim więc chwycisz młotek do ręki, zastanów się czy koprocesor jest ci przydatny.

Procesor główny a koprocesor.

Procesor główny naszej Amigi 1200 to MC68020. Jest on unowocześnioną wersją pradziada z Amigi 500/600/2000 - procesora MC68000. Jego główne zalety to znacznie szybsza praca, więcej instrukcji oraz interesujący szczegół - pełna współpraca z koprocesorami Motoroli. Są to układy MC68881/68882 - drugi z nich jest prawie dwukrotnie szybszy od pierwotnego 68881.

Procesor główny ma dość ograniczone możliwości. Wykonywane przez niego działania to podstawowe dodawanie, odejmowanie, dzielenie, mnożenie, negowanie (zmiana znaku). Posiada instrukcje operujące na danych logicznych (bity i pola bitowe), oraz szereg instrukcji przesyłania danych i sterowania programem. Jego zaletą jest ogromna szybkość wykonywania operacji (MC68020 do 10 mln operacji na sek.), wadą - dość niewygodny format liczb. Są one:
- ograniczone zakresem <-2147483648> - <+2147483647>,
- stałoprzecinkowe. Procesor nie uwzględnia istnienia ułamków. Najszybsza operacja dzielenia ASR uznaje, że 5/2 = 2 (!!!), 9/2 = 4, itd. Bardziej zaawansowana instrukcja DIV jako dodatkową informację podaję resztę z dzielenia.

Z podanych przykładów wynika, że stworzenie programu korzystającego z bardziej zaawansowanych danych (np. zwykłego kalkulatora) jest dość złożone oraz nieefektywne. Aby ułatwić tego typu obliczenia organizacja IEEE opracowała "uczony" format zmiennoprzecinkowy. W formacie tym każda liczba składa się z trzech składników:

$$\text{Liczba} = \text{mantysa} * \text{podstawa}^{\text{wykładnik}}$$

Podstawa ma stałą wartość, dla systemów powszechnego użytku (np. 10 dla kalkulatorów, nie mylić z PeCetami, które niektórzy na przekór ich możliwości zaliczają do gromady komputerów). Komputery (Amigi i inne stacje robocze) korzystają z systemu dwójkowego, stąd podstawa ma wartość 2.

Oto kompletny format liczby IEEE:
- Mantysa = z zakresu 1 .. 2
- Podstawa = 2
- Wykładnik = liczba dwójkowa
- Znak (ujemny/dodatni) dla mantysy i wykładnika

$$\text{Liczba} = 1.xxx * 2^{yyy}$$

Motorolka z naszej A1200 nie potrafi bezpośrednio liczyć na tak złożonych danych, jeśli program korzysta z danych zmiennoprzecinkowych czas jego wykonania dąży do nieskończoności.

Specjalnie dla liczb zmiennoprzecinkowych wymyślono koprocesory matematyczne (ang. Floating Point Unit), które potrafią wykonywać szereg działań arytmetycznych oraz dodatkowo funkcje trygonometryczne np. Sin(x). Procesor nie musi liczyć na danych, do których nie jest przygotowany, stąd wykonanie programu jest znacznie szybsze.

Z koprocesora korzystają wszelkiego typu programy inżynieryjne, matematyczne, wyższej klasy graficzne np. Real3D, ImageFX, Cinemorph, itp. Wykorzystanie tych ostatnich na 2 MB pamięci jest niemożliwe, jednak programy generujące fraktale, DTP oraz inne mniej "pamięciożerne" zaczną pracować znacznie szybciej na komputerze wykorzystującym koprocesor.

Późniejsze dodanie pamięci pozwoli na pełne użytkowanie tych programów.

Instalujemy koprocesor.

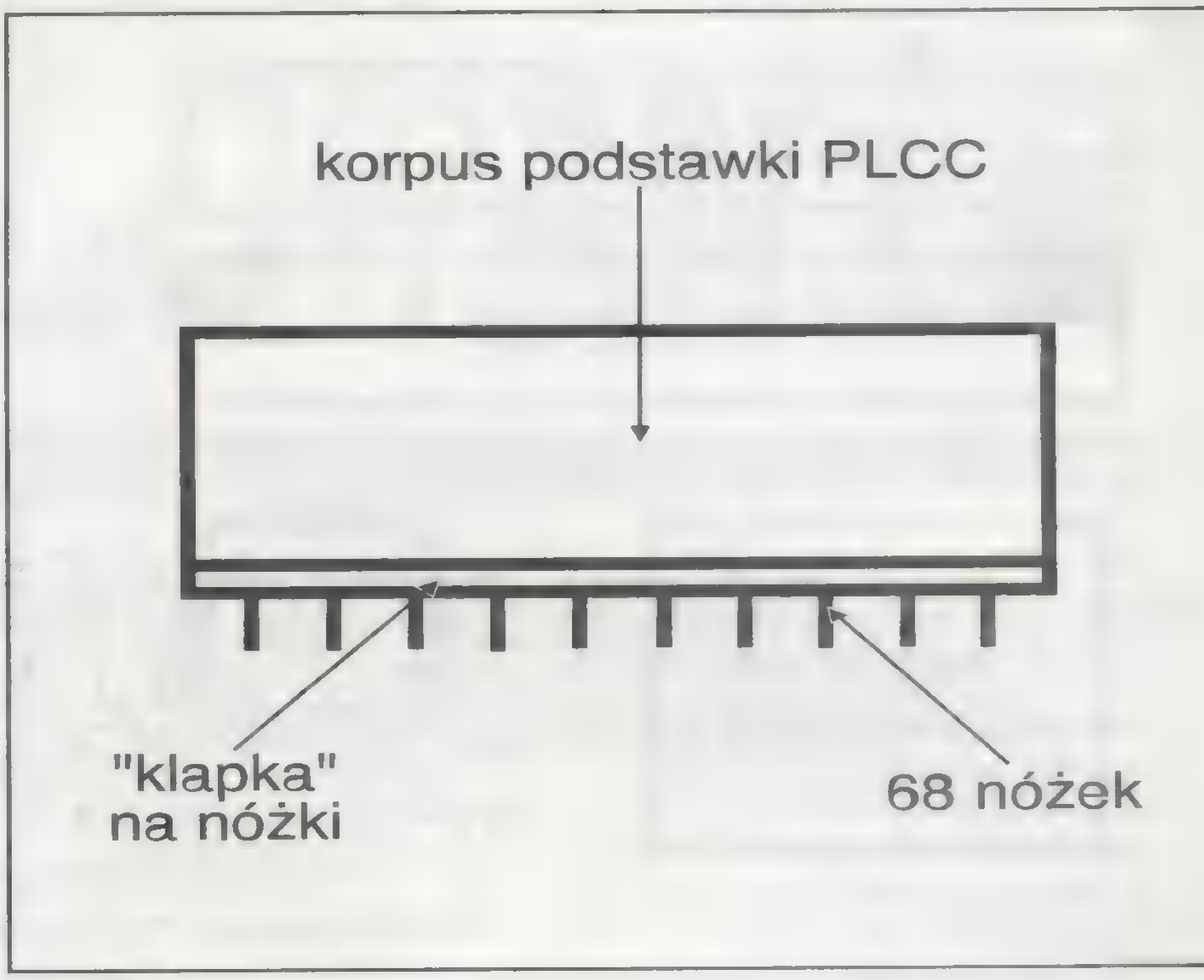
W komputerach firmy Commodore dodanie koprocesora jest maksymalnie utrudnione. Amiga 1200 jest wykonana w technologii montażu powierzchniowego, stąd zarówno instalacja, jak i kupno koprocesora jest dość trudne. Koprocesor MC68881 jest zgodny elektronicznie i mechanicznie z nowszym MC68882. Oba wykonywane są w dwóch wersjach:

- MC 68881/68882RC - obudowa PGA
- MC 68881/68882FN - obudowa PLCC

Obudowa PGA jest praktycznie niemożliwa do wykorzystania w naszej A1200. Musimy więc zaopatrzyć się w MC68881/82FN.

Musimy także wybrać rodzaj koprocesora. Wersja MC68882 jest prawie dwukrotnie szybsza od MC68881. Programy specjalnie wykorzystujące jej możliwości są szybsze nawet czterokrotnie. Koprocesor MC68881 jest niewiele tańszy, więc zalecam kupno tego drugiego. Pozostaje wybór odpowiedniego zegara.

Motorola produkuje koprocesory taktowane zegarem 16, 20, 25, 33, 40, 50MHz. Koprocesor współpracuje z procesorem głównym w zupełnie asynchroniczny sposób - częstotliwości zegara obu układów mogą być zupełnie różne. W standardowych A1200 z fabrycznie montowanymi FPU częstotliwość zegara wynosi 14,4 MHz dla obu układów. Jeśli nie zamierzamy dokonywać żadnych zmian na płycie głównej, to nasz koprocesor będzie taktowany właśnie takim zegarem bez względu na jego faktyczne możliwości. Najprostsza instalacja to kupno MC68882FN 16 MHz i zainstalowanie go bez dodatkowych zabiegów. Jak wcześniej wspomniałem, będzie on taktowany zegarem 14,4 MHz.



Rysunek 1.

Jeśli bardzo zależy Ci na prędkości koprocesora, musisz przeciąć ścieżkę na płycie głównej, dolutować kabelek i wybrać jedno z poniższych rozwiązań:

- Kupić MC68882 25 MHz i taktować go systemowym zegarem 28,8 MHz. Drobna nadwyżka w prędkości nie zaszkodzi w pracy koprocesora. Instalacja jest równie prosta jak w przypadku 14,4 MHz (1 kabelek).
- Kupić dowolny wybrany przez siebie koprocesor (najlepiej 50 MHz) i taktować go zewnętrznym zegarem. Instalacja jest nieco bardziej skomplikowana od pozostałych. Dodatkowo podłączamy zasilanie do zegara (łącznie 3 kabelki). Szczególnie polecam jednoukładowe zegary, które są podłączane do zasilania +5V i masy, dając wyjście TTL do bezpośredniego połączenia z koprocesorem. Zegar taki można kupić w dowolnym sklepie elektronicznym lub u dealera Motoroli (adres na końcu artykułu).



Rysunek 2.



Rysunek 3.

Po wybraniu odpowiedniego typu połączenia musimy kupić:

- Koprocesor. Powinien być przystosowany do podstawki PLCC lub montażu powierzchniowego.

- Podstawkę PLCC z 68 nogami. Powinna mieć dwa "wianuszki" nóżek na dolnej części. Koniecznie sprawdź, czy jest ona rozbieralna na dwie części! Można również kupić podstawkę PLCC w wersji przystosowanej do montażu powierzchniowego.

- Zegar, tylko jeśli zdecydujesz się na częstotliwość inną niż 14,4 lub 28,8 MHz.

- Klej przewodzący, jeśli masz podstawkę do montażu powierzchniowego.

Zakupioną podstawkę musimy rozebrać zdejmując dolną część. *(Patrz Rysunek 1.)*

Teraz nogi z wewnętrznego rzędu układamy równolegle do zewnętrznych (zakładam, że potrafisz posługiwać się prostymi narzędziami), drobnymi nożyczkami obcinamy nie potrzebną część nóżek uzyskując pojedynczy wianuszek "nóg" o długości kilku milimetrów. *(Patrz Rysunek 2.)*

Po obejrzeniu miejsca na podstawkę, będziesz musiał odpowiednio skorygować ich długość.

Otwieramy obudowę.

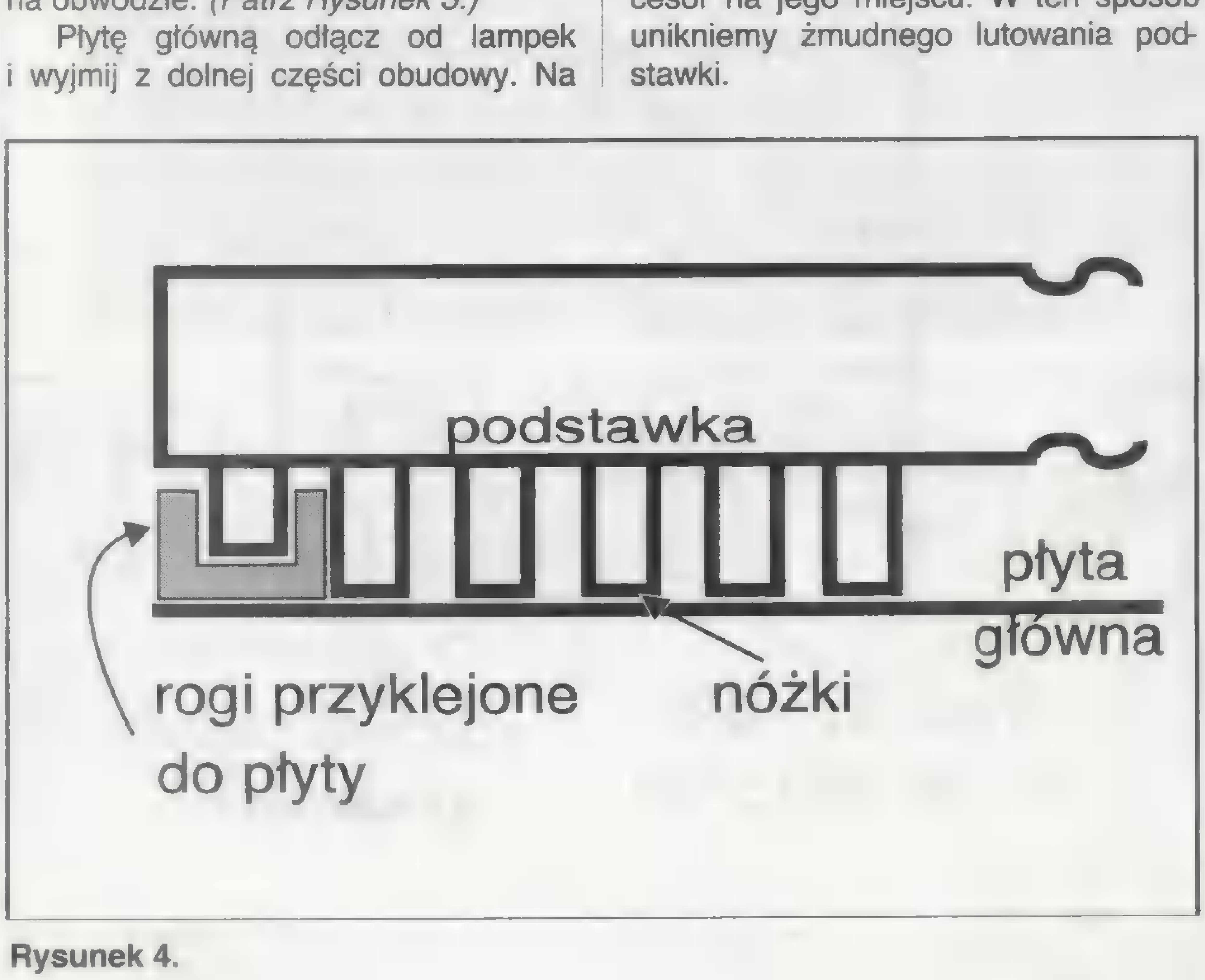
Pięć śrubek na spodniej części obudowy, plomba gwarancyjna i jesteśmy w środku. Delikatnie zdejmij szarą klamrę na HD (jeśli posiadasz A1200 HD musisz najpierw odłączyć kabel łączący twardy dysk z płytą główną), odkręć i odłącz stację dysków, a następnie zdejmij srebrzysty ekran chroniący płytę główną. Odszukaj procesor główny MC 68EC020 (jest to jeden z najmniejszych układów w Amidze!), obok niego znajduje się miejsce pod koprocesor - kwadrat ze stykami na obwodzie. *(Patrz Rysunek 3.)*

Płytę główną odłącz od lampek i wyjmij z dolnej części obudowy. Na jednym z boków podstawka będzie miała trójkącik lub wgłębienie (często będzie on (ono) na wewnętrznej części podstawki, później przykryty koprocesorem). Bok ten powinien pokrywać się z zaznaczoną na płycie białą jedynką. Delikatnie dopasuj podstawkę, zwracając uwagę na kontakt wszyskich nóg do odpowiednich styków.

Montaż koprocesora 14,4 MHz.

Zaznacz położenie podstawki. Wcześniej odcięte rogi z dolnej części podstawki przyklej w odpowiednich miejscach płyty głównej. Rogi "klapki" mają dziurki w które wejdą odpowiednie wypustki podstawki. Tym samym klejem kontaktowym przyklej podstawkę do płyty głównej - przyklejaj ją do rogów. Nogi powinny być odgięte od podstawki tak, aby po przyklejeniu stykały się z odpowiednimi kontaktami na płycie. Do podstawki włóż koprocesor i zamknij obudowę komputera. *(Patrz Rysunek 4.)*

Można również bezpośrednio przykleić koprocesor, co jest nawet łatwiejsze od montażu podstawki. Gdy nóżki koprocesora są zagięte pod jego korpus, można zarówno wstawić go w podstawkę ze stykami bocznymi, jak i przykleić spodnią częścią. Każdą nóżkę należy delikatnie posmarować klejem przewodzącym (uważając, aby nie doszło do zalania klejem dwóch lub więcej nóg!) i spokojnie położyć koprocesor na jego miejscu. W ten sposób unikniemy żmudnego lutowania podstawki.

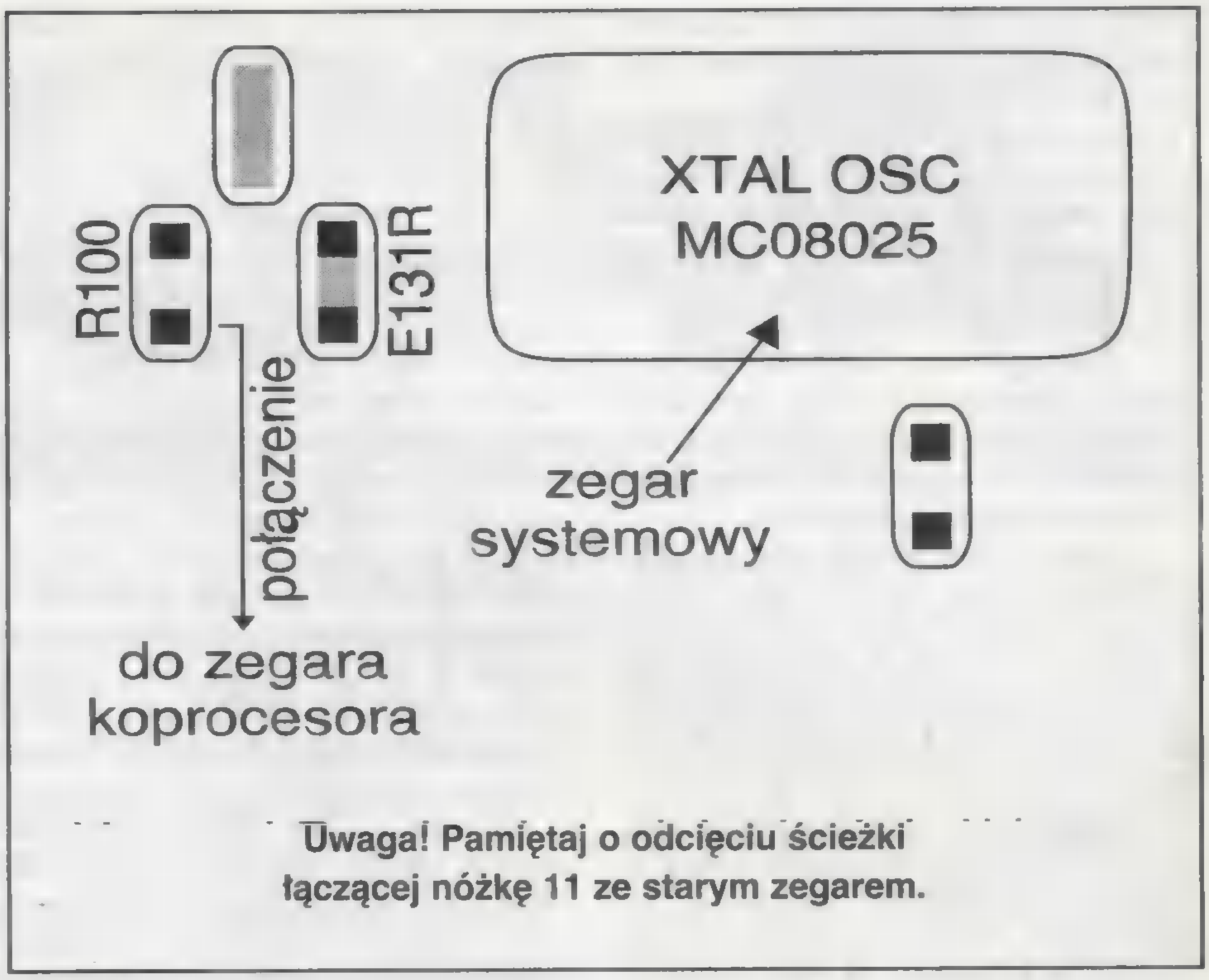


Rysunek 4.

Montaż koprocesora 28,8 MHz.

Zanim przykleisz podstawkę, musisz od niej delikatnie oderwać nóżkę, która doprowadzała oryginalny zegar 14,4 MHz - doprowadzenie nr 11. Jeśli przyklejasz sam koprocesor, przetnij ścieżkę prowadzącą do tej nogi. Do górnej części tej nóżki przylutuj kabelek. Odszukaj metalowy układ, który jest zegarem systemowym dla całego komputera (przypomina on miniaturkę puszki po sardynkach). *(Patrz Rysunek 5.)*

Kabelek połączony z końcówką zegara (łącze z oderwaną nóżką) przylutuj do zaznaczonej części kondensatora. W większości komputerów jest tam tylko styk, gdyż wcześniej montowany kondesatorek okazał się zbędny. Przyklej podstawkę i zamknij obudowę.



Rysunek 5.

Montaż koprocesora z dowolnie wybranym zegarem.

Tak jak w przypadku zegara 28,8 MHz musisz zewnętrznie doprowadzić zegar odcinając nóżkę nr 11. Kabelek podłączony do tego styku łączymy z wybranym przez nas zegarem, zapewniając mu połączenie z zasilaniem i masą. Zegar musi być jednoukładowym generatorem odpowiedniej częstotliwości, dającym odpowiedni sygnał TTL. Najlepiej nadają się do tego układy z metalową obudową podobne do zegara systemowego. Wystarczy wtedy wybrać odpowiednią częstotliwość, nie dbając o symbol czy producenta.

Do zegara dodatkowo podłączamy kondensatorek elektrolityczny. Zapobiega on zakłóceniom wytwarzanym przez zegar. *(Patrz Rysunek 6.)*

Nóżka nr 1 jest oznaczona kropką. Po przylutowaniu kabli do zegara, przyklej go do wewnętrznej części ekranu. Ostrożnie zamknij ekran i upewnij się, że zegar nie dotyka żadnych elementów na płycie głównej. Teraz zamknij obudowę.



Rysunek 6.

Test na obecność FPU.

Po zainstalowaniu koprocesora musisz sprawdzić reakcję systemu. Uruchom program ShowConfig z dysku Extra:Tools. Obok napisu "processor" program powinien wyświetlić FPU 68881 lub 68882. Jego obecność można również wykryć programami testującymi prędkość komputera np. AIBB czy SysInfo.

Teraz sprawdź czy zainstalowany procesor poprawnie liczy. Uruchom Calculator z Workbench3.0:Utilities. Jeśli pracuje on poprawnie i nie podaje wyników typu 2+2=5, możesz odsapnąć - Twój komputer ma poprawnie zainstalowany koprocesor. Jeśli komputer nie rozpoznaje koprocesora lub pracuje niepoprawnie, spróbuj poprawić położenie podstawki, sprawdź podłączenie zegara.

Interfejs MC 680xx z MC 6888x

Dla tych, którzy już zainstalowali koprocesor lub mają zamiar go zainstalować podam teraz odrobinę teorii ko

nicznej do zrozumienia jego działania oraz pozwalającej uniknąć kolizji z innymi procesorami (karty turbo).

Motorola opracowała specjalny system komunikacji procesora ze specjalnymi urządzeniami nazwanymi koprocesorami. System Motoroli jest jedynym połączeniem procesora z koprocesorem, które:

- jest w pełni asynchroniczne. Procesor i koprocesor mogą pracować z dowolnymi prędkościami. Cykle przesyłania danych są potwierdzane w chwili gotowości danego urządzenia.

- zakłada stosowanie maksymalnie 8 koprocesorów, każdy z nich może wybrać dowolny numer. Komunikacja z koprocesorem o danym numerze jest realizowana programowo, stąd program korzystający z koprocesora musi znać jego numer.

- koprocesor nie ma ustalonego charakteru pracy ani typów danych. Jego zasoby - rejestry i instrukcje - są "dodawane" do modelu programowego procesora MC680xx. W chwili obecnej istnieją: koprocesory realizujące ochronę zasobów procesora - MMU (ang. Memory Management Unit), koprocesory matematyczne FPU (ang. Floating Point Unit) oraz szereg koprocesorów sieciowych.

Przesłanie rozkazu do koprocesora jest czynnością złożoną. Żeby Was nie nudzić, podaję uproszczony przebieg działania:

1. Procesor rozpoznaje specjalne słowo instrukcji o formacie $Fxxx, które oznacza cykl współpracy z koprocesorem.

2. Na podstawie danych zawartych w tym słowie procesor rozpoznaje numer koprocesora 0-7 oraz cykl dostępu.

3. Kolejne długie słowo jest przesyłane do koprocesora. Jest to instrukcja koprocesora. Teraz koprocesor decyduje o ewentualnym pobraniu danych z zewnątrz - z rejestrów procesora lub z pamięci.

4. Jeśli procesora nie ma, program jest przerywany i następuje interwencja systemu operacyjnego (w przypadku standardowego systemu Amigi, ratunek kończy się "wysypaniem" systemu i ramką Guru).

Punkty, które wypisałem są realizowane jedynie na procesorze MC68020 lub nowszym. Procesory starsze: MC 68008-12 nie realizują takich cykli automatycznie przerywając program (stan wyjątkowy linii A i F). Podłączenie koprocesora do tych procesorów wymaga drobnych zabiegów "kosmetycznych" w systemie operacyjnym. Musi on emulować cykle dostępu do procesora w chwili przerwania programu.

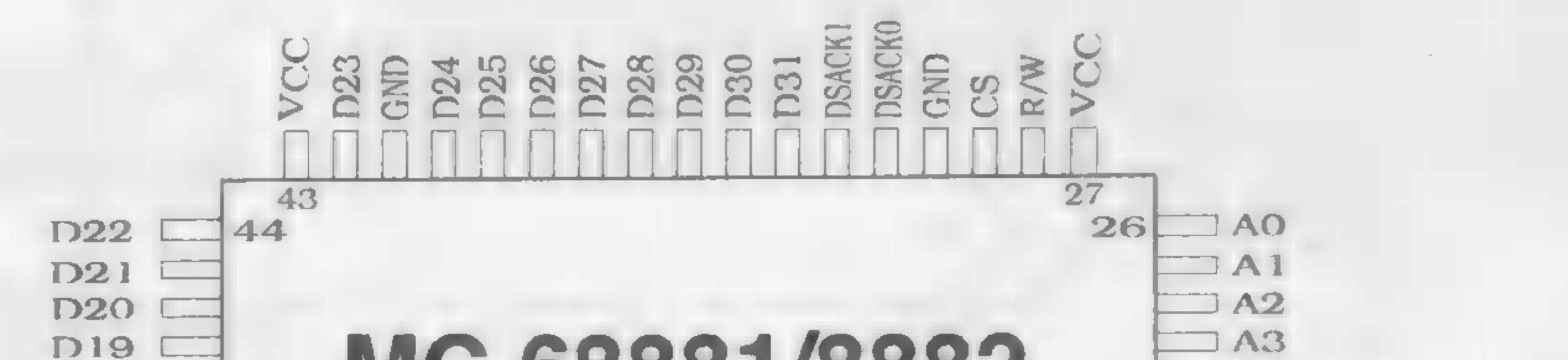
Zanim przeanalizujemy współpracę koprocesora z procesorami nowszymi niż MC68020 garść dodatkowych informacji. Większość procesorów 32 bitowych współpracuje ze specjalnymi układami MMU. Pośredniczą one w dostępie do pamięci i innych urządzeń, zapewniając ochronę niektórych obszarów przestrzeni adresowej (pamięć + układy IO) przed dostępem programów. Układ ten pomaga w tworzeniu pamięci wirtualnej i maszyny wirtualnej (żadna z tych cech nie jest używana w stanadardowej Amidze). Układ MMU według konwencji Motoroli jest koprocesorem (MC68051) i jest dołączany z zewnątrz tak jak FPU MC68881/82. Jego numer jest ustalony na 0 (nie ma możliwości użycia innego koprocesora jako nr 0, jak i układu MMU jako koprocesora nr 1-7). W nowszych procesorach (MC68030/40/60) układ ten jest zintegrowany z procesorem.

Układ MC68881/82 może mieć numer 1-7 (0 zarezerwowane dla MMU). W Amidze 1200, 2500 i 3000 ma on numer 1, zgodnie z sugestiami Motoroli (nie jest to konieczność). Tak samo jest z wszelkiej maści kartami Turbo posiadającymi układ FPU. Wprawdzie podłączenie karty z wbudowanym koprocesorem do Amigi 1200 z FPU na płycie nie zakończy się "Big bangiem", lecz niepoprawne funkcjonowanie komputera jest pewne (Jeśli mimo to Twój komputer będzie pracował poprawnie, koniecznie skontaktuj się z serwisem, prosząc o usunięcie usterki - poprawnego działania). Pozostaje wyjęcie wolniejszego koprocesora i jego odsprzedanie.

Jeśli do swojej Amigi 1200 zamierzasz kupić kartę z procesorem MC68040 lub MC68060 (karty takie chwilowo nie istnieją), to na pewno nastąpi kolizja. Procesory te zawierają procesory FPU wewnątrz struktury (ma on numer 1 tak jak FPU na płycie). Dostęp do procesora na płycie w ogóle nie nastąpi, zostanie przechwycony przez wewnętrzny FPU. Tutaj też doradzam sprzedaż zamontowanego procesora.

Dołączenie karty turbo bez zamontowanego koprocesora, z procesorem MC 68020/30 lub MC 68040EC (uproszczona wersja MC 68040) nie koliduje z koprocesorem na płycie. Procesory te będą wykorzystywać nasz koprocesor.

Myślę, że na dzisiaj wystarczy Wam wiadomości o koprocesorze. Za miesiąc dalszy ciąg naszych rozważań. ❑



Rysunek 7. Wprowadzanie układu MC68881/68882.

PORÓWNANIE ODZYSKIWACZY

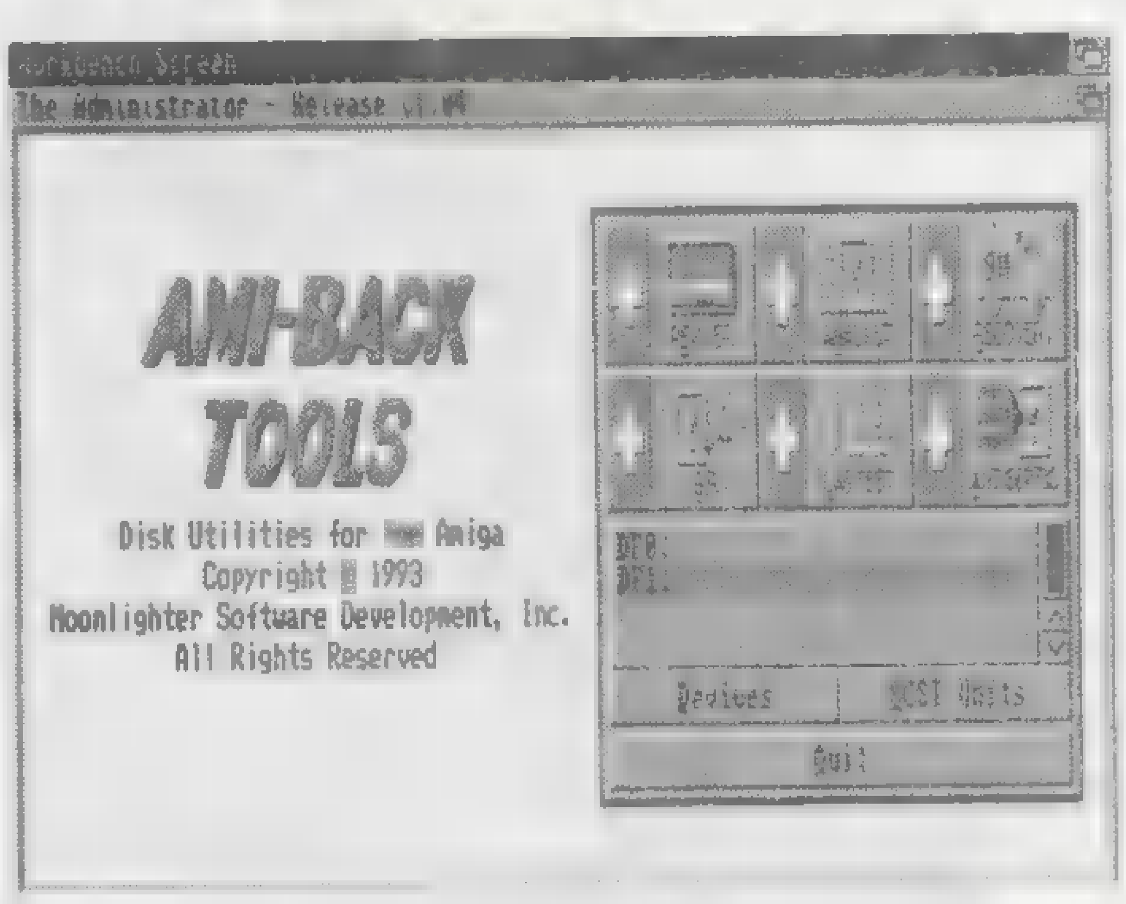
Dane jakie gromadzimy na dyskietkach lub dysku twardym są nieraz rzeczą niezwykle cenną, ze względu na swoją wartość materialną, pracę włożoną w ich przygotowanie, bądź wagę informacji (np. bazy danych różnych firm). Każdy stara się więc zabezpieczyć je jak najlepiej. Pełne bezpieczeństwo jest jednak nieosiągalne, a dążenie do niego niezwykle kosztowne.

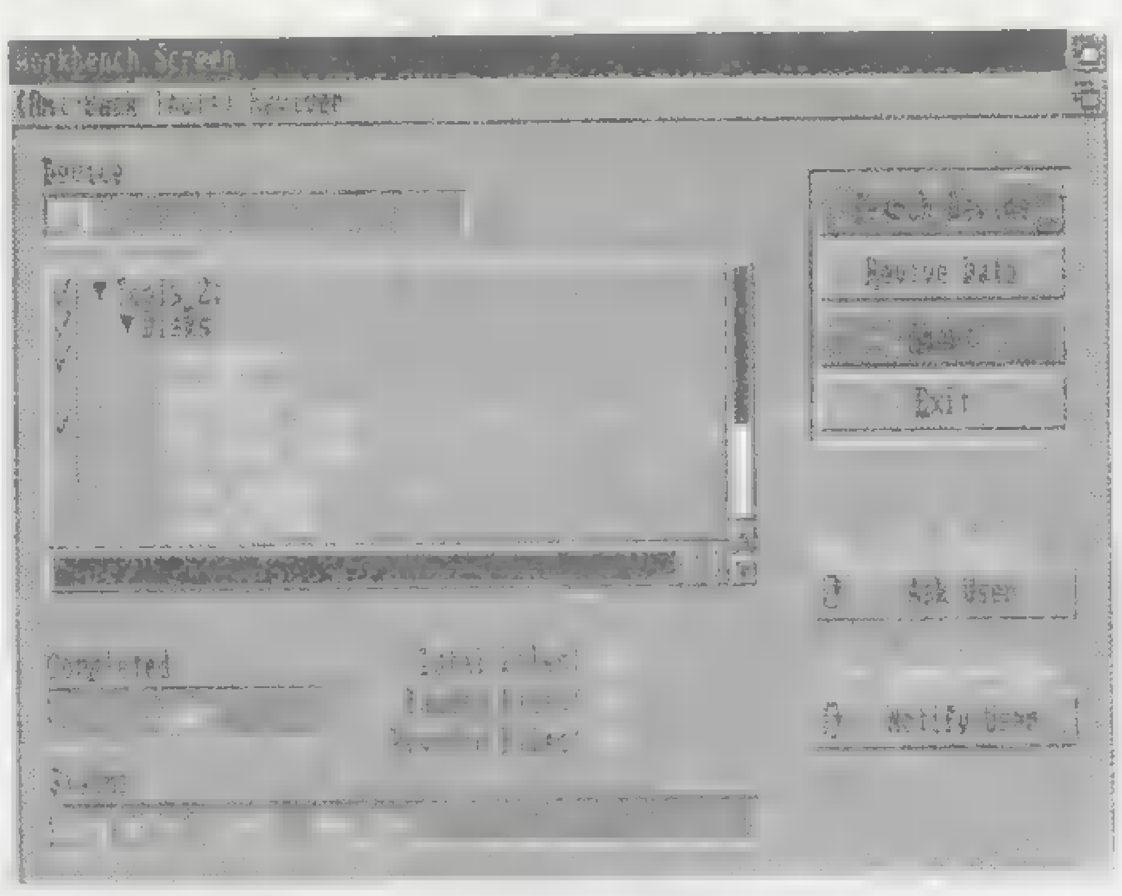
Zdarza się zatem, że pieczołowicie zbierane dane giną w wyniku awarii sprzętu, ataku wirusa lub czyjegoś niedbalstwa. Trzeba wtedy sięgnąć po program przeznaczony do odzyskiwania uszkodzonych plików. QUARTERBACK TOOLS i DISKSALV to już prawie standardy w tej dziedzinie, a z kolei AMI-BACK TOOLS jest jednym z nowszych i mniej znanych narzędzi tego typu. Z tego też względu zajmę się nim w pierwszej kolejności.

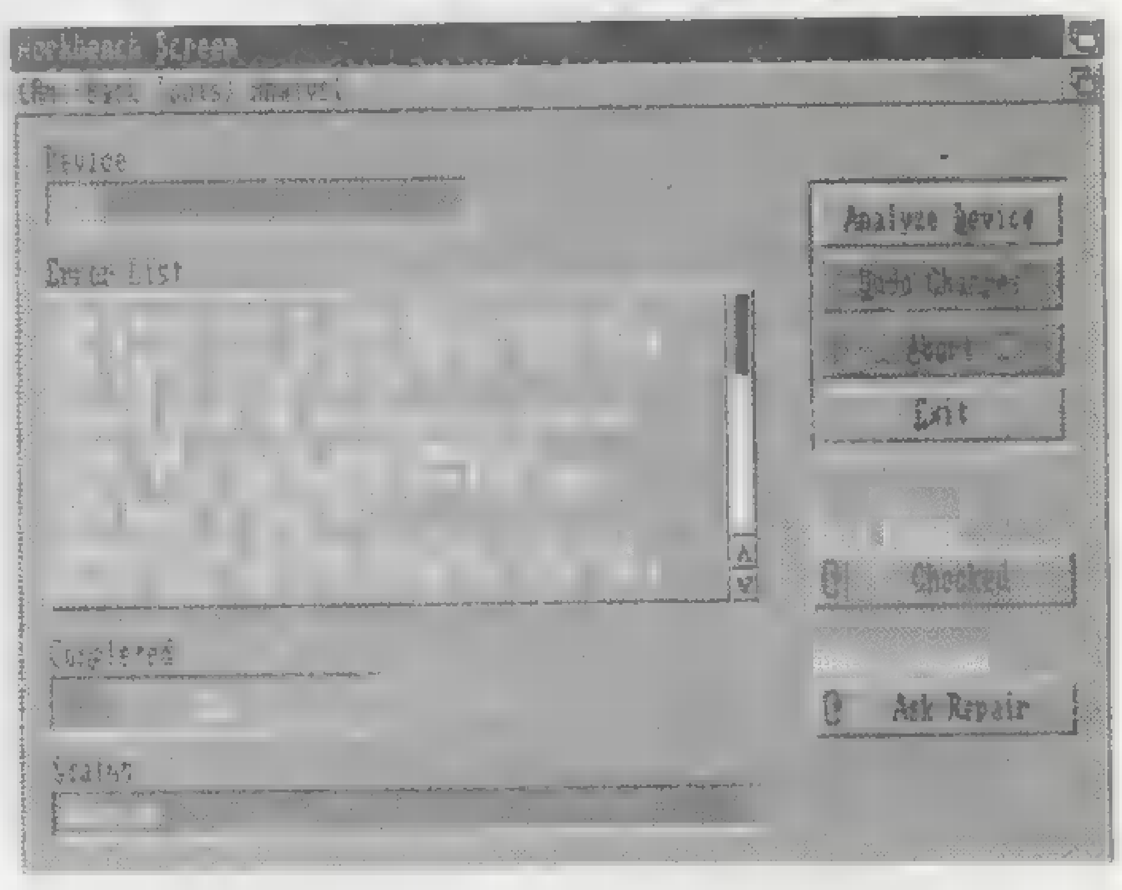
Różnie to bywa z nowościami, lecz w tym wypadku program wart jest uwagi, ze względu na funkcjonalność i pewność działania. Po jego uruchomieniu użytkownik ma do dyspozycji prawie wszystko co jest potrzebne do ratowania uszkodzonych danych i to podane w bardzo elegancki sposób. Ekran główny programu przedstawia się raczej skromnie i nie zapowiada sporej ilości dostępnych opcji. Widzimy na nim wypisany wielkimi literami tytuł, informacje o producencie, wersji itp. oraz w prawej części spis urządzeń zainstalowanych w systemie. Nad nim sześć ikon, za którami kryje się cała moc programu. Reprezentują one sześć ekranów pomocniczych pozwalających kontrolować większość jego funkcji.

Na ekranie głównym jest też standardowe menu "z belki" zawierające kilka pomocnych, lecz rzadko używanych opcji. Przeznaczone są one do ustawienia preferencji, czyli kolorów ekranu, dokonania wyboru czy program ma się uruchamiać na własnym screenie czy w oknie Workbencha itp. Są to opcje raczej standardowe w programach na AMIGĘ i nie będę się nad nimi rozwodził. Dodam tylko, że z menu można wywołać wszystkie ekrany pomocnicze (jeśli ktoś nie lubi klikania na ikony). Czas teraz na przedstawienie konkretnych możliwości opisywanego programu, a więc ekrany pomocnicze.

Na pierwszy ogień pójdzie Reviver, funkcja dla roztargnionych. Pozwala ona na odzyskanie plików przypadkowo skasowanych z dysku przez użytkownika. Nie jest to wcale tak rzadki przypadek jak mogłoby się wydawać. Zdarza się, że ktoś kasuje zbiór, uznając go za niepotrzebny, a później okazuje się, iż był w błędzie. O ile nie posiada on tzw. "kopii bezpieczeństwa" musi uciekać się do takich programów. Odzyskanie skasowanych zbiorów jest bardzo proste. Wystarczy uruchomić przeszukiwanie wybranego wcześniej na ekranie głównym urządzenia i po chwili ujrzymy spis możliwych do odzyskania plików. Po podjęciu decyzji (ustawieniu parametrów) dotyczących postępowania programu w przypadku plików nie możliwych do odczytania i pojawiania się okienka żądającego potwierdzenia przy każdym odtwarzanym zbiorze, można uruchomić funkcję Revive Data. Po tej operacji pliki ponownie znajdą się na dyskietce lub dysku twardym. Jak wcześniej wspomniałem, w ten sposób można odzyskać tylko zbiory przypadkowo skasowane. Odradzam natomiast, stosowanie funkcji Reviver do odtwarzania plików uszkodzonych w innych oko



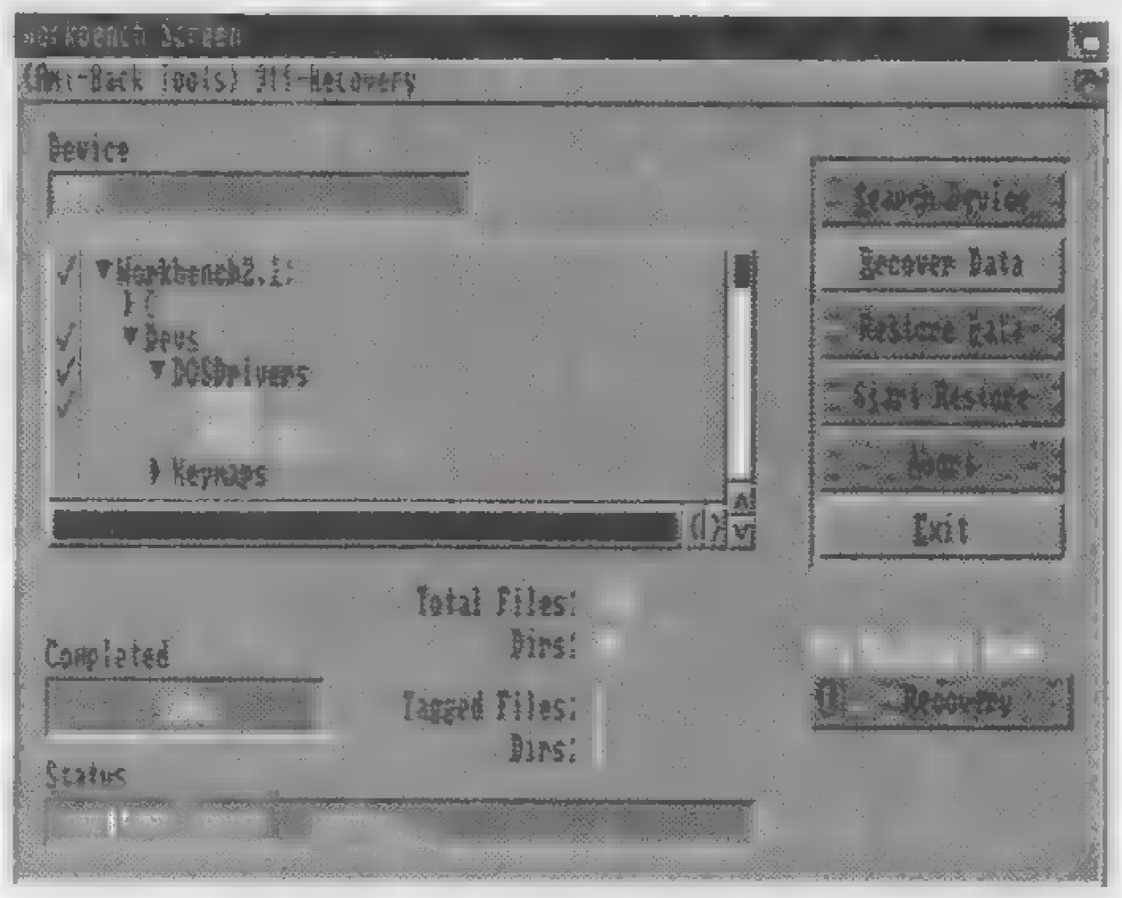




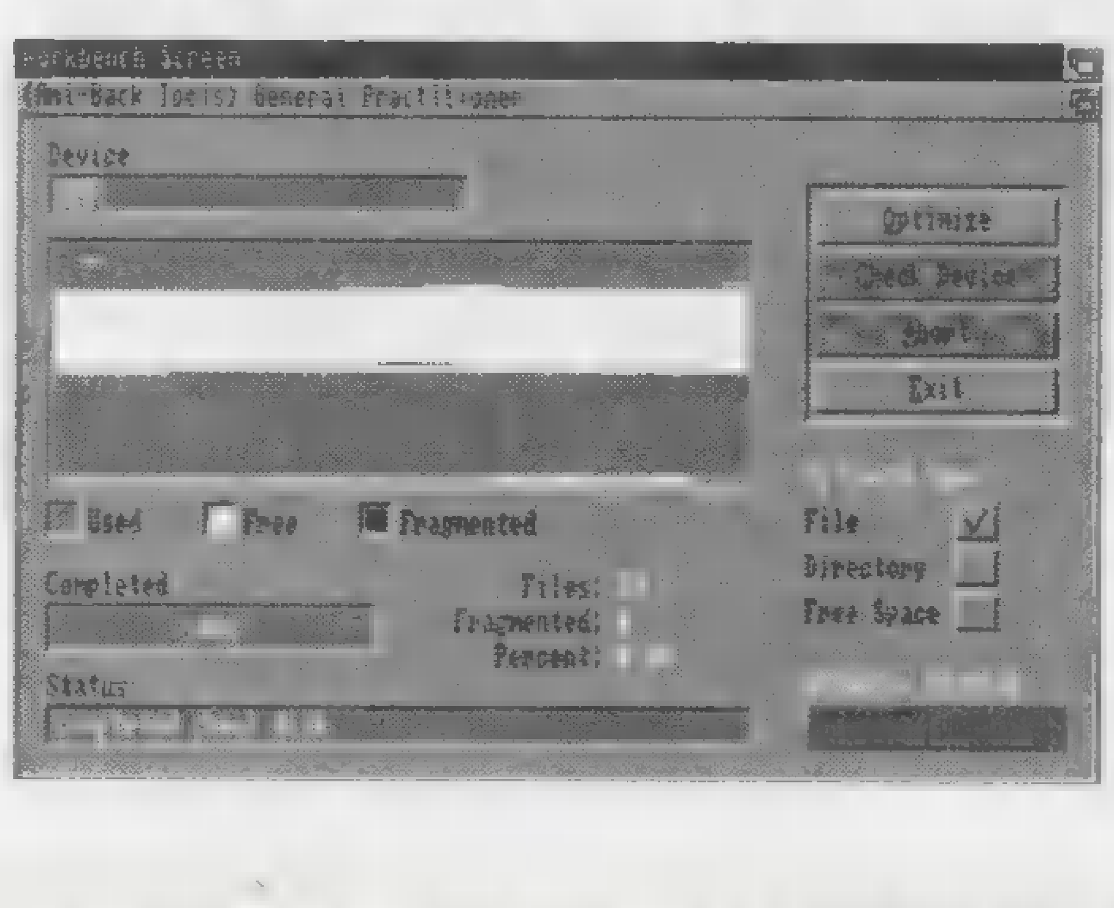
licznościach, gdyż odtworzony plik jest zapisywany bezpośrednio na przeszukiwanym dysku. Przy uszkodzonej strukturze systemowej próba zapisu może pogorszyć sytuację. W takim przypadku należy zastosować inną funkcję programu AMI-BACK TOOLS, ale o tym w dalszej części. Podsumowując, z plikami skasowanymi Reviver radzi sobie doskonale i można go stosować bez obaw.

Kolejny ekran pomocniczy wywołamy uruchamiając opcję Analyst. Przeznaczona jest ona do sprawdzania i naprawy struktury dysku, uszkodzonej w przypadku niewłaściwego działania jakiegoś programu. Uszkodzenia pliku Disk-validator np. w wyniku działania wirusa rówież są przedmiotem działania tej funkcji programu.

Jeśli więc mamy dysk z uszkodzoną strukturą, wystarczy uruchomić tę funkcję i wszystko wróci do normy. Owszem, w większości przypadków. Ale może się też zdarzyć, że naprawa skończy się niepowodzeniem i biorąc pod uwagę fakt zapisywania bezpośrednio na naprawianym dysku, dane możemy stracić bezpowrotnie. Trzeba więc zabezpieczyć się przez wykonanie wspomnianej już kopii bezpieczeństwa. Służy do tego funkcja 911-Recovery pozwalająca na odzyskanie z danego dysku "wszystkiego co się da" i przekopiowanie tego materiału na inne urządzenie (druga stacja dysków, ramdysk, inna partycja itp.). W przypadku dyskietek, prościej będzie chyba wykonać kopię programem X-Copy, ale przy dysku twardym, opcja 911-Recovery będzie raczej niezastąpiona. Pozwala ona na wykonanie backupu "twardziela" przy uszkodzonej walidacji, z czym inne programy wykonujące tę operację mogą sobie nie poradzić. W tym miejscu ktoś może zapytać, do czego jest w zasadzie funkcja Analyst, skoro można pliki odzyskać i przekopiować na inny dysk, a ten uszkodzony sformatować i dane przekopiować ponownie. Można, ale ile to trwa? Jest to operacja możliwa w przypadku dyskietki, lecz przy dysku twardym trwałoby to strasznie długo. Z tego względu, lepiej jest wykonać kopię i uruchomić funkcję Analyst. Jeśli się powiedzie, możemy zaoszczędzić sobie połowę pracy, a jeśli nie, to pozostaje formatowanie i kopiowanie. Gdy mamy już kopię, możemy przystąpić do naprawy dysku. Po ustawieniu parametrów dotyczących sprawdzania bloków danych, klikamy na Analize Device i czekamy na wynik mając nadzieję, że będzie on pozytywny.

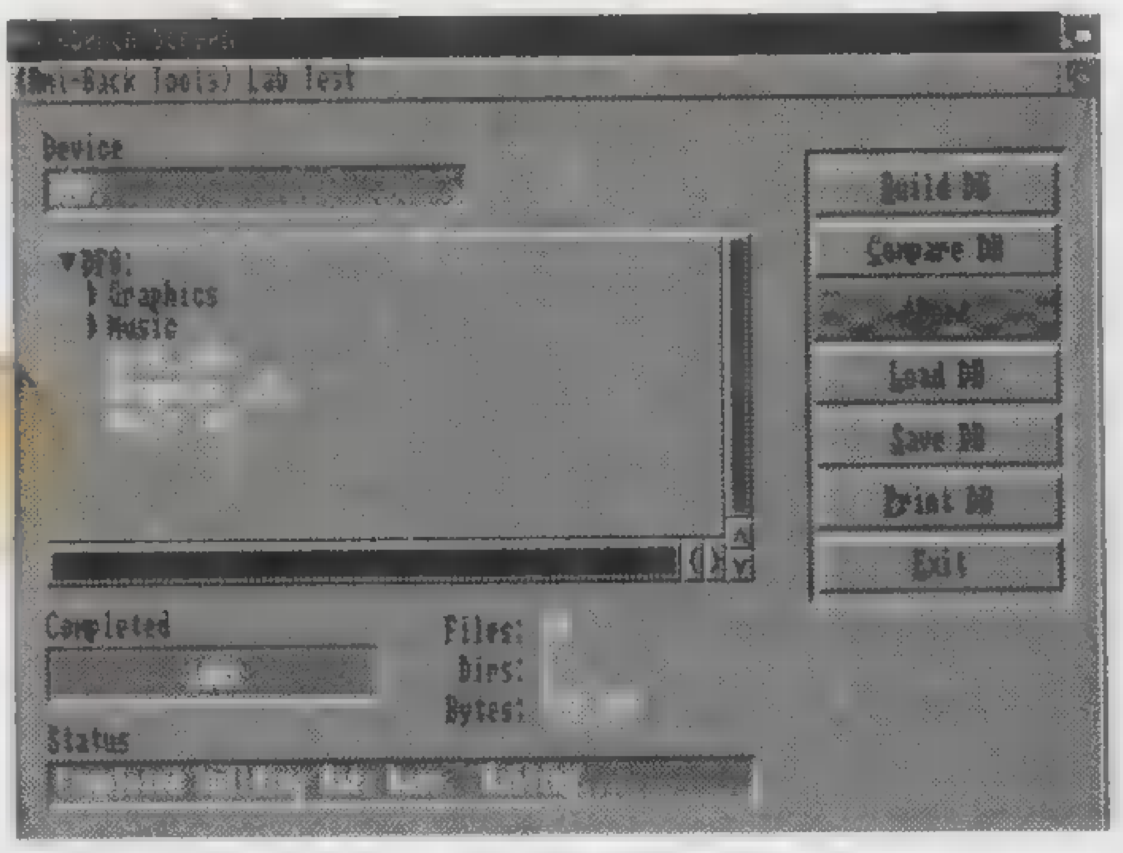


Teraz kilka zdań na temat archiwizacji uszkodzonego dysku. Po uruchomieniu funkcji 911-Recovery, ukazuje się odpowiedni ekran pomocniczy. Pierwszą rzeczą którą trzeba zrobić, to odpowiednie ustawienie opcji 911-Recovery Mode, a tym samym ustalenie, czy będziemy dokonywać archiwizacji, czy odtwarzać zarchiwizowane dane na sformatowany dysk. W pierwszym przypadku przeszukiwanie dysku zakończy się wyświetleniem wszystkich możliwych do odczytania plików. Następnie należy oznaczyć te, które mają zostać skopiowane i uruchomić archiwizację. Pojawi się jeszcze ekran z informacjami na temat ilości plików, ich łącznej długości itd. oraz menu pozwalające wybrać urządzenie docelowe przy kopiowaniu i włączyć automatyczną kompresję danych. Jeśli backup będziemy wykonywać na dyskietki, to stosownie do pojawiających się komunikatów, trzeba będzie wkładać je do wybranej stacji. Zapisane zostaną w sposób niestandardowy i odtworzenie danych będzie możliwe tylko przy pomocy funkcji 911-Recovery, po odpowiednim ustawieniu 911-Recovery Mode (na Restore). Operacja ta przebiega podobnie jak archiwizacja z tą tylko różnicą, że dane są odczytywane z niestandardowo zapisanych dyskietek i następnie zapisywane na wybranym dysku już jako pliki w standardowej postaci. Trwa to niestety dosyć długo i szczerze współczuję wszystkim, którzy będą zmuszeni do ratowania w ten sposób dużego "twardziela". Czasami jednak będzie to jedyne wyjście i pozostanie uzbroić się w cierpliwość.

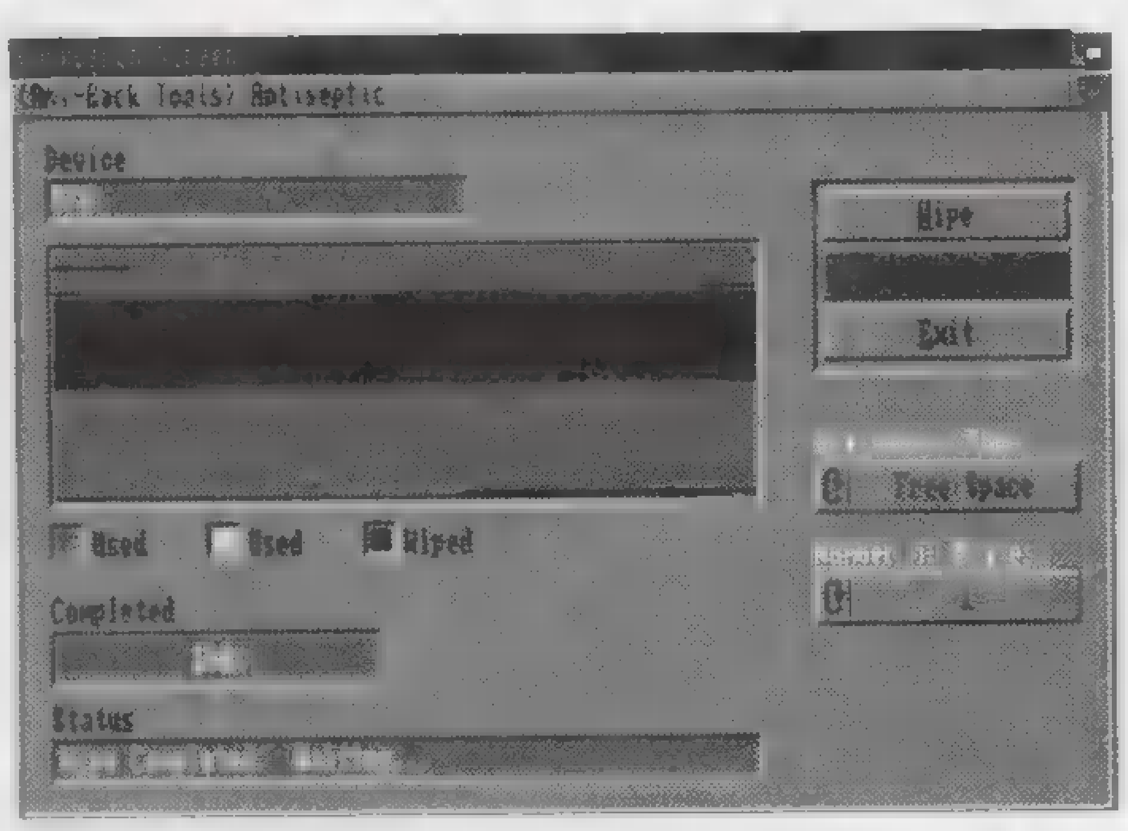


Jeśli już mowa o czasie oczekiwania, to wypada napisać o kolejnej możliwości programu AMI-BACK TOOLS jaką jest optymalizacja dysków. Jak zapewne wszystkim wiadomo, odpowiednie poukładanie danych na dysku przyspiesza ich wczytywanie. Do takiego "układania" czyli optymalizacji, przeznaczona jest funkcja General Practitioner (w skrócie GP). Przed uruchomieniem tej operacji, można ustawić jeszcze kilka parametrów mających duży wpływ na jej przebieg. Oznaczenie ("ptaszek") parametru File powoduje układanie danych należących do jednego pliku w blokach następujących kolejno po sobie. Najlepsze wyniki uzyskuje się w przypadku długich plików, gdzie wzrost prędkości wczytywania jest wyraźny. Wskazane częste stosowanie. Następny parametr to Directory. Jak łatwo się domyśleć powoduje on ułożenie danych tak, aby katalog dysku wczytywany był jak najszybciej. Rzeczywiście, w programie Directory Opus (lub podobnym) katalog tak zoptymalizowanej dyskietki wczytywany jest bardzo szybko. Pojawił się pewien problem, gdy chciałem wykonać "dir" tej dyskietki pod Workbenchem. Ikony pojawiały się znacznie wolniej!? niż przed optymalizacją. Po kilku próbach doszedłem do wniosku, że odpowiedzialny za to jest algorytm działania programu optymalizującego "dopasowujący" ułożenie tzw. "nagłówków" do odczytywania listy plików. Workbench natomiast czyta "nagłówek" pliku, a następnie plik z definicją jego ikony, aby ją wyświetlić. Powoduje to ciągłe przesuwanie się głowicy stacji i stratę czasu. Kolejny parametr to Free Space powodujący zgrupowanie wszystkich danych zawartych na dysku w jednym ciągłym bloku. Jako wynik otrzymamy wolne miejsce na reorganizowanym urządzeniu także w postaci jednego ciągłego bloku. Opcję tę warto stosować dla wszystkich rodzajów dysków. Po jej wykonaniu, dogranie nawet kilkunastu zbiorów powoduje tylko nieznaczny spa

dek prędkości wczytywania. Pozostaje tylko decyzja w sprawie graficznej prezentacji przebiegu operacji i można kliknąć na OPTIMIZE uruchamiając proces.

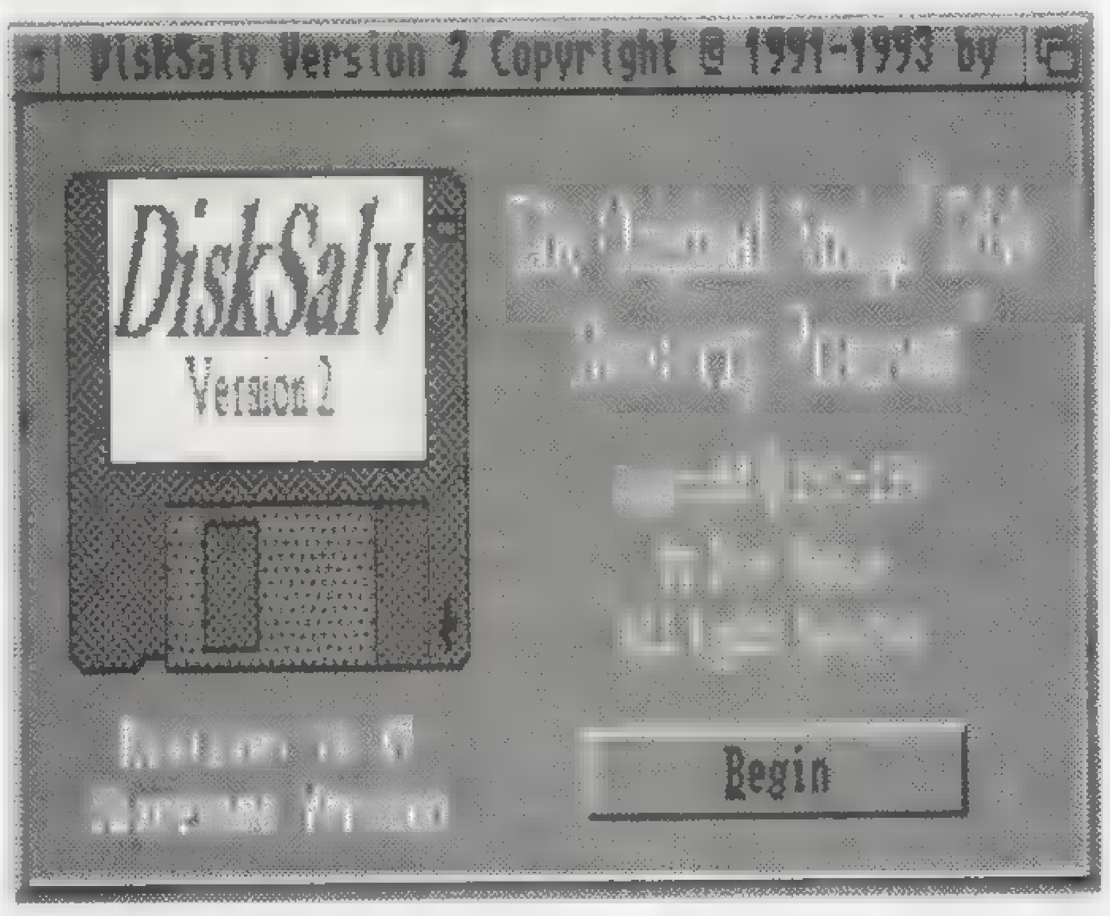


Funkcja Lab Test przeznaczona jest do wykonywania spisu wszystkich plików danego dysku. Rejestracja danych przeprowadzana jest automatycznie. Program odczytuje "directory" wraz ze wszystkimi informacjami dodatkowymi dotyczącymi danego zbioru (data powstania, długość, protekcje itd.). Następnie tworzy z nich własny plik, który można nagrać na dysku lub wydrukować. Możliwe jest także uzupełnianie tak stworzonej bazy danych i oczywiście odtwarzanie jej w razie potrzeby. Podstawowym zastosowaniem tej operacji jest wykonanie spisu plików dysku poddawanego archiwizacji, gdyż po jakimś czasie mogą powstać problemy z zapamiętaniem zawartości kilkunastu dyskietek leżących gdzieś w pudełku.

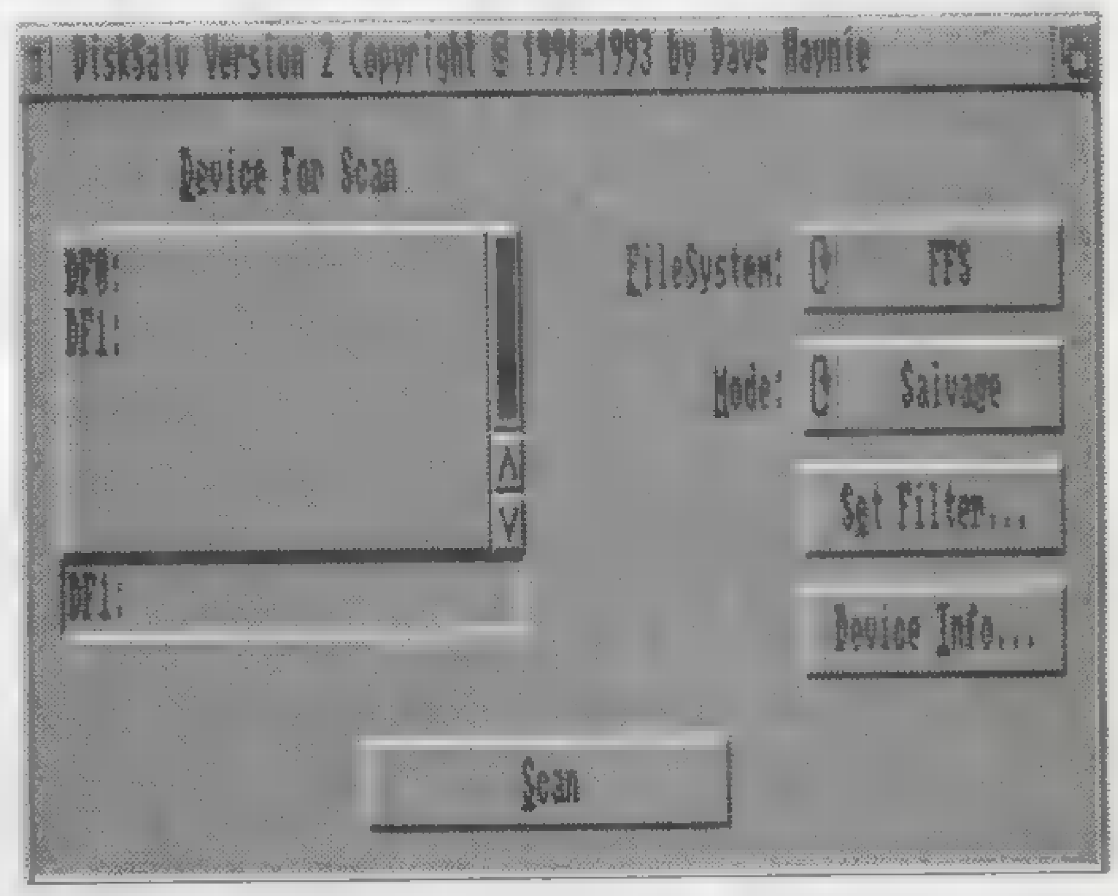


Opcje dostępne na ostatnim ekranie pomocniczym Antiseptic przeznaczone są, ogólnie mówiąc, do całkowitego kasowania danych. Dziwić może takie działanie programu, bądź co bądź przeznaczonego do odzyskiwania danych. Zdarza się jednak tak, że trzeba wyczyścić dyskietkę całkowicie i program ten robi to doskonale. Program posiada (ciekawą) możliwość wyczyszczenia także takich obszarów dysku, na których nie są zapisane żadne dane. Skutkiem działania takiego czyszczenia jest usunięcie wszelkich pozostałości plików skasowanych i uszkodzonych.

Po tej sporej dawce informacji na temat AMI-BACK TOOLS czas na porównanie jego możliwości ze wspomnianymi na początku "renomowanymi" programami. Przejdę więc do programu DISKSALV.

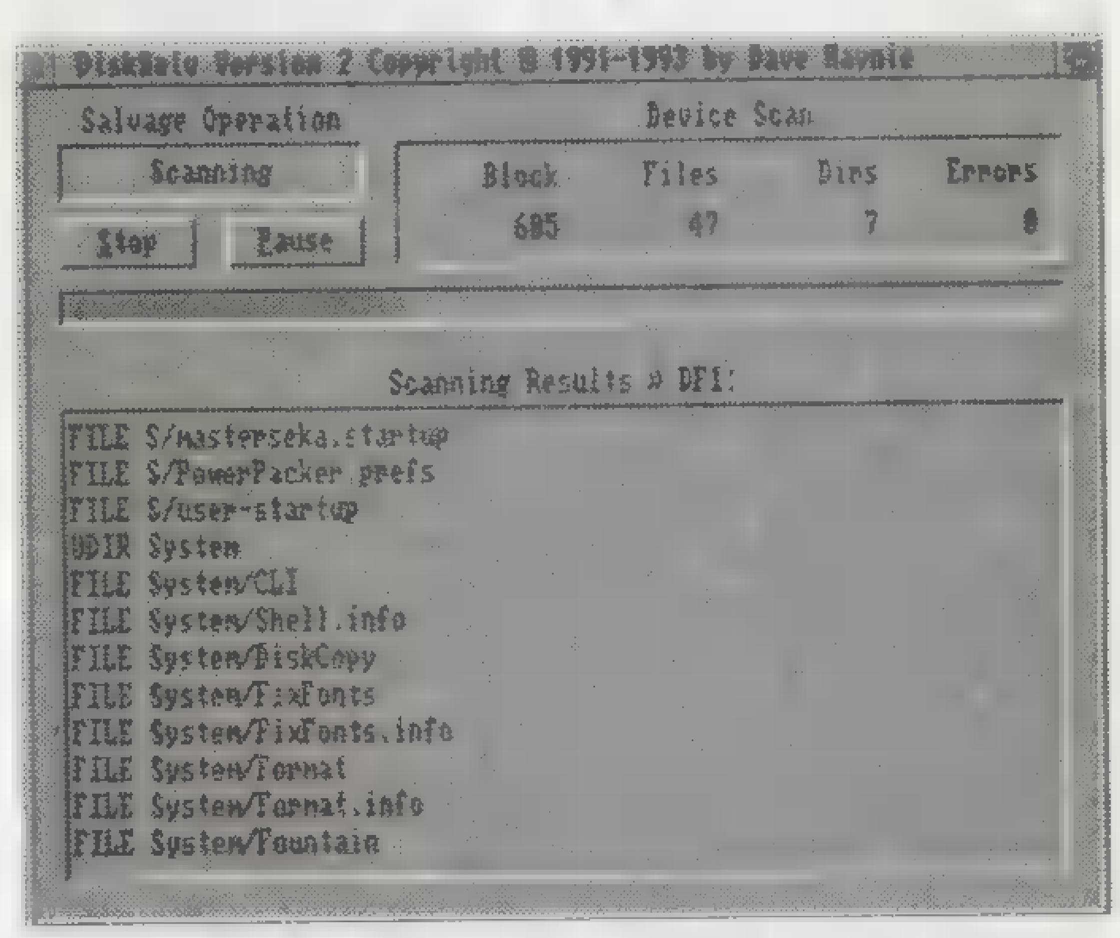


Na pierwszy rzut oka wygląda on znacznie skromniej od pozostałych. Otwiera na ekranie Workbencha jedynie własne okno ze spisem dostępnych urządzeń i kilkoma gadżetami. Mamy też do dyspozycji dwa menu "z belki" Settings i Project. Pierwsze zawiera kilka opcji związanych z ustawieniem parametrów pracy całego programu, a w drugim oprócz typowych raczej opcji About, Help i Quit można znaleźć jeszcze kilka funkcji pomocniczych, do których wrócę w dalszej części artykułu. Pierwszy z gadżetów widocznych na ekranie przeznaczony jest do ustawienia standardu zapisu danych na dysku, z którym chcemy pracować. Dostępne są wszystkie występujące na Amidze rodzaje zapisu, a więc i te najnowsze wprowadzone z systemem 3.0. Sporym ułatwieniem jest automatyczne rozpoznawanie standardu po wybraniu właściwego urządzenia. Dzięki temu gadżet Filesystem przyjmuje właściwie rolę informacyjną i ręczne ustawianie standardu zapisu nie jest konieczne. Kolejny "przycisk" to "Mode" czyli wybór trybu pracy programu. Do wyboru mamy pięć opcji: Salvage, Undelete, Validate, Repair i Unformat, do których wrócę za chwilę. "Set Filter" pozwala na ustalenie parametrów skanowania dysku (pojawia się nowe okno). Możemy włączyć "filtrację" ze względu na ustawienie bitów protekcji odczytywanych plików oraz zadeklarowany "File Pattern" także zawężyć obszar przeszukiwania dysku do określonego ciągu bloków. Device Info, jak się pewnie wszyscy domyślają, podaje informacje na temat wybranego urządzenia (w dodatkowym okienku). Ostatni gadżet to Scan, uruchamia on proces ustawiony w opcji "Mode".



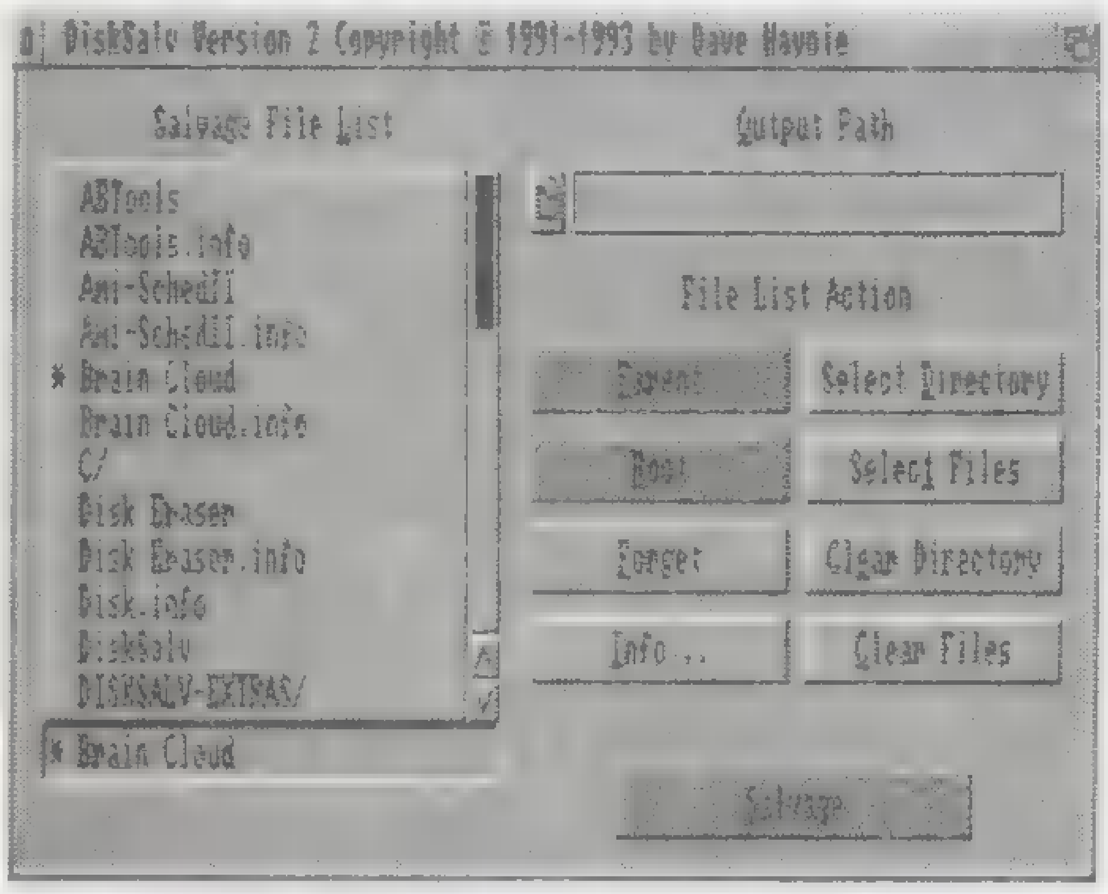
Jak widać opcja ta jest bardzo ważna, gdyż zależnie od jej ustawienia program wykonuje zupełnie inne operacje. Nie sposób oprzeć się wrażeniu, że w zależności od wybranego trybu pracujemy z innym programem. Składa się na to wiele czynników, lecz najistotniejszym jest chyba odmienny wygląd okien pomocniczych pojawiających się podczas pracy lub zmiany menu "z belki". Postaram się więc krótko przedstawić pracę programu na pięć wariantów, stosownie do ustawionego trybu.

Salvage



Jest to tryb najczęściej używany i chyba najbardziej uniwersalny. Ustawia się on automatycznie po wczytaniu programu i do jego uruchomienia wystarczy kliknąć na Scan (oczywiście po uprzednim wybraniu urządzenia). Po chwili na ekranie monitora pojawi się nowe okno przedstawiające przebieg operacji. Można na nim znaleźć wiele interesujących informacji. W dolnej części na bieżąco wyświetlane są tytuły plików i katalogów odczytywanych z dysku. Zarówno tych zapisanych poprawnie, jak i odzyskanych. W górnej części możemy znaleźć numer aktualnie od

czytywanego bloku dysku, liczbę odnalezionych plików i katalogów oraz pod pozycją Errors liczbę błędów. Stopień zaawansowania operacji obrazowany jest typowym "paskiem" zwiększającym swą długość w miarę upływu czasu. Korzystając z gadżetów Stop i Pause możemy przerwać, bądź zatrzymać na chwilę wykonywanie operacji. Wraz z pojawieniem się nowego okna uzyskujemy dostęp do nowego menu " z belki", lecz w tym wypadku ma ono jedynie dwie opcje Help i Quit, których działania nie muszę chyba wyjaśniać. Chociaż proces odczytywania dysku przebiega z umiarkowaną prędkością, to w przypadku partycji dysku twardego o dużej pojemności konieczna będzie odrobina cierpliwości. W tym miejscu trzeba jednak przyznać, że operacja przebiega nieco szybciej niż podobna w programie AMI-BACK TOOLS (911-Recovery).



Prędzej czy później skanowanie dobiegnie końca i na ekranie zobaczymy nowe okno. W lewej jego części znajduje się typowy requester przeznaczony do wyboru plików. Ich oznaczenie i poruszanie się po katalogach przebiega w sposób jak najbardziej standardowy (przez kliknięcie na wybranej nazwie), a opcje Parent, Root i Forget działają tak jak w wielu innych programach. Pewną innowacją jest funkcja Info podająca informację o ostatnio wskazanym pliku lub katalogu. Bardzo wygodne okazały się także funkcje Select Directory i Select Files. Pierwsza powoduje automatyczne zaznaczenie wszystkich katalogów i plików w nich zawartych, ■ druga oznacza pliki "widoczne bezpośrednio" (z pominięciem tych zawartych w katalogach). Do anulowania tak wprowadzonych oznaczeń przeznaczone są odpowiednio opcje Clear Directory i Clear Files. Jako ciekawostkę mogę podać, że wśród plików odnalezionych na dysku zawsze pojawia się katalog Disksalv-Extras zawierający dwa pliki: Boot-Block, w którym zawarte są dane odczytane z bootblocku przeszukiwanego dysku i Volume-Info zawierający nazwę dysku i datę zapisu.

Po zaznaczeniu wszystkich interesujących nas plików trzeba podać miejsce, w które mają zostać przekopiowane. Ścieżkę dostępu do katalogu lub pliku należy wpisać w polu oznaczonym jako Output Path. Można także kliknąć na gadżet znajdujący się na lewo od tego pola i ustalić ścieżkę przy pomocy standardowego requestra. Dalszy przbieg operacji zależy od oznaczenia opcji Archive Stream w menu Settings (z belki). Jeśli opcja ta nie zostanie odznaczona, to wybrane pliki przekopiowane zostaną do podanego w Output Path katalogu w postaci standardowej. Jeśli natomiast oznaczymy tę opcję, to zostaną one zarchiwizowane i nagrane we wskazanym katalogu jako jeden duży plik. Dla poprawnego działania tej funkcji wymagane jest podanie pełnej ścieżki dostępu, nie tylko do katalogu, ale i tytułu pliku, jaki ma zostać utworzony. Wspomniane menu Settings zawiera także kilka opcji pomocniczych pozwalających na ustawienie parametrów archiwizacji (archiwizacja dat powstania plików, bitów protekcji, komentarzy itp.). Kopiowanie uruchamia się przez kliknięcie na Salvage. Odtworzenie plików zarchiwizowanych możliwe jest jedynie przy pomocy funkcji Restore z menu Project, dostępnego przy aktywnym oknie głównym programu. Działanie tego trybu jest analogiczne do 911-Recovery w AMI-BACKu, choć brakuje tu możliwości bezpośredniej archiwizacji dużej ilości danych na dyskietki.

Undelete

Tym razem opis będzie bardzo krótki, gdyż tryb ten działa bardzo podobnie jak poprzedni (Salvage). Wykorzystuje te same okna i menu. Jedyną różnicą jest to, że archiwizację i kopiowanie możemy stosować jedynie do plików odtworzonych. Te, które były zapisane poprawnie pozostają ukryte. Tryb ten jest najczęściej używany do odzyskiwania plików przypadkowo skasowanych. Opcja jest całkowicie bezpiecznia, ponieważ automatycznie kopiuje odzyskane dane na inne urządzenie.

Validate

Tryb ten przeznaczony jest do naprawy walidacji dysku. Działa on dosyć szybko, lecz może być nieskuteczny w przypadku poważniejszych uszkodzeń. Daje natomiast dobre wyniki przy dyskach uszkodzonych przed niewłaściwe zadziałanie programu Disk-Validator. Jest to więc tryb podobny w działaniu do funkcji Analyst programu AMI-BACK TOOLS (patrz wyżej).

Repair

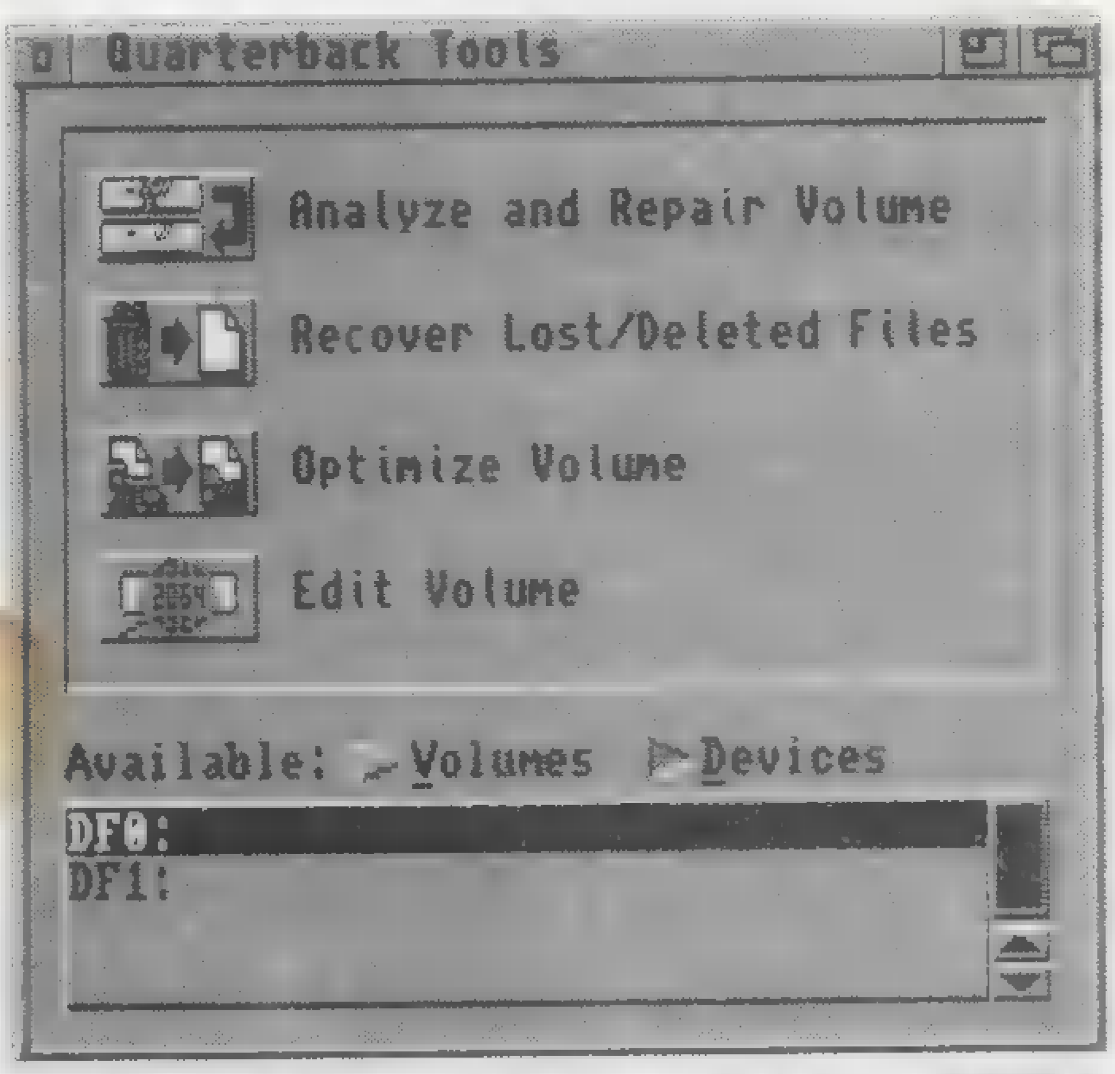
Tryb ten działa w zasadzie tak samo jak Validate, lecz znacznie wolniej i co za tym idzie dokładniej. Radzi sobie zarówno z naprawą walidacji, jak i innymi uszkodzeniami struktury dysku. Zalecany jest do stosowania w "cięższych przypadkach". Podobnie jak poprzedni nie wymaga w zasadzie ustawiania parametrów, a jedynie potwierdzenia zamiaru wykonania operacji po pojawieniu się ostrzeżenia o możliwości "pogorszenia sytuacji" (zapis bezpośrednio na dysk).

Unformat

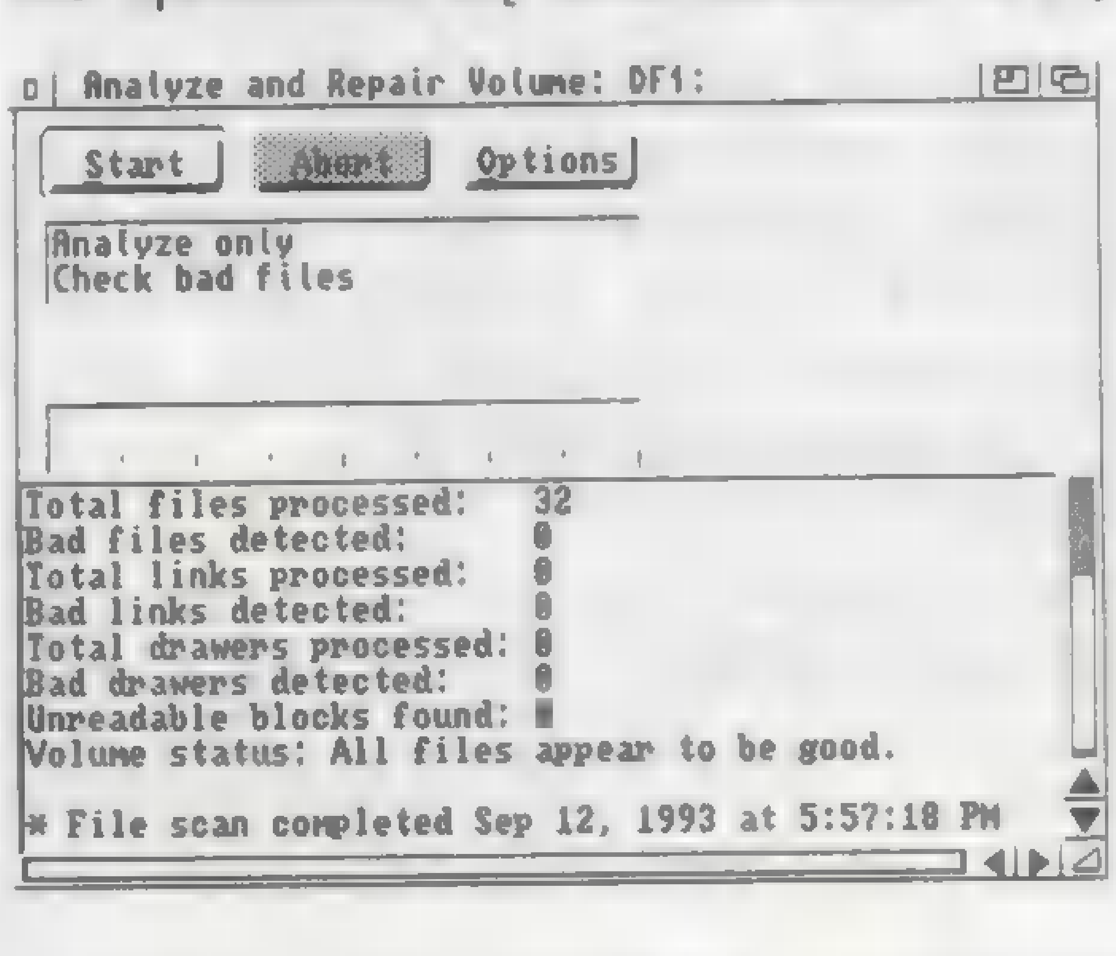
Nazwa trybu jest niesłychanie imponująca, lecz jak się z pewnością wszyscy domyślają, nie do końca prawdziwa. Tryb ten pozwala na odtworzenie danych, ale tylko po tzw. szybkim formatowaniu, które nie usuwa wszystkich danych z dysku. Wykonanie "prawdziwego" formatowania jest jednoznaczne z całkowitą utratą danych i żaden program nie jest w stanie odwrócić tego procesu. Także i tutaj nie jest konieczne ustawianie żadnych parametrów, a informacje podawane są w oknie identycznym jak przy innych trybach (np. Salvage).

Na zakończenie opisu DISKSALVa pragnę zauważyć, że mimo iż liczba opcji dostępnych w tym programie jest mniejsza niż w AMI-BACK TOOLS, to nie brakuje niczego istotnego. Wynika to raczej z różnicy w idei obu programów. AMI-BACK pomyślany jest jako "kombajn" do wszystkiego, a DISKSALV jest programem specjalizowanym. Wyłącznie do odzyskiwania uszkodzonych danych. Uważam nawet, że jest to plus DISKSALVa, który wydaje się być prostszy w obsłudze. Na koniec pozostawiłem informacje o nowej wersji pro

gramu, który stał się już historią "świata AMIGI" czyli QUARTERBACK TOOLS.



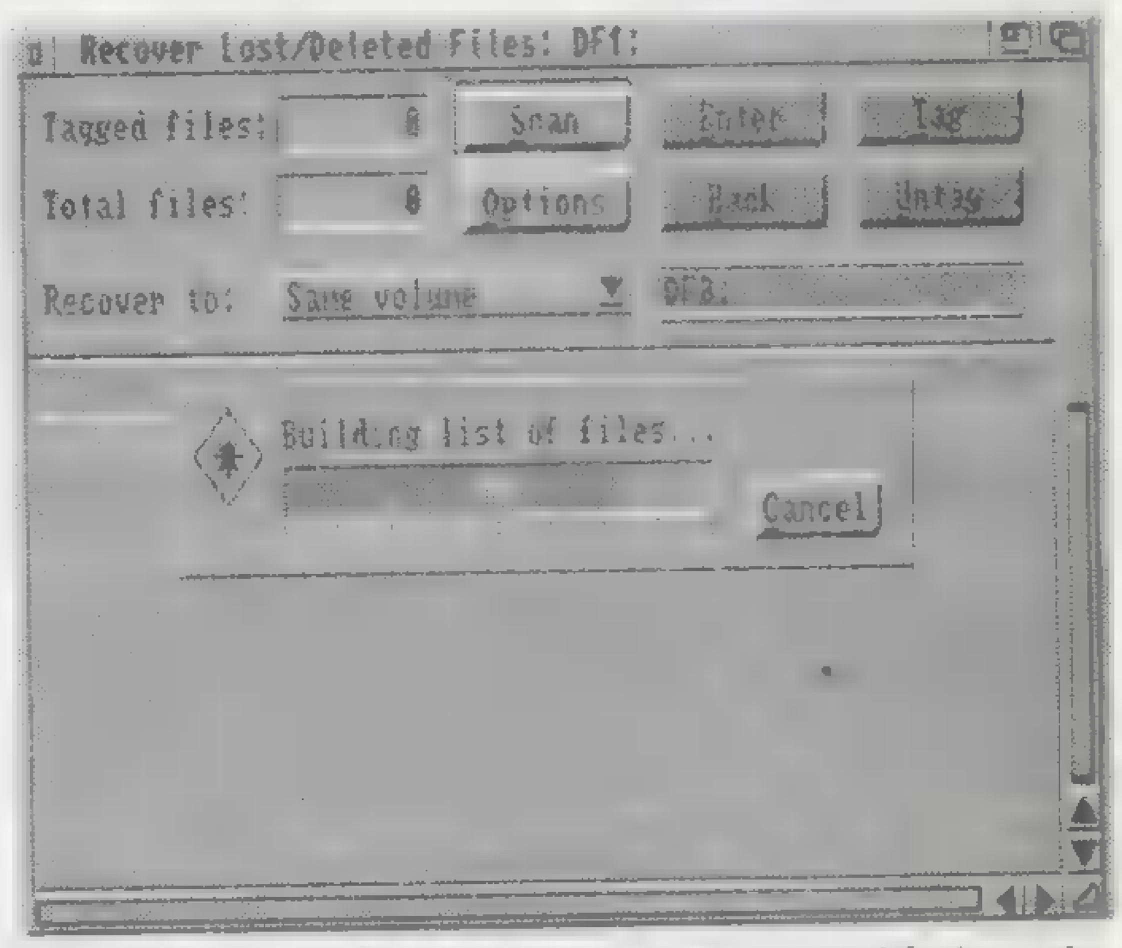
Podobnie jak w przypadku DISKSALVa QUARTERBACK TOOLS otwiera na ekranie Workbencha jedynie niewielkie okno. W jego dolnej części znajduje się spis urządzeń dostępnych w systemie. Cztery duże ikony widoczne w oknie służą do wywołania okien pomocniczych, przeznaczonych do obsługi czterech podstawowych bloków funkcyjnych programu. Dostępne jest także menu "z belki". Jest ono identyczne dla wszystkich bloków programu i zawiera tylko opcje podstawowe, podobnie jak to miało miejsce w AMI-BACK TOOLS. Znaleźć w nich można opcję Preferences ustawiającą podstawowe parametry pracy całego programu, cztery opcje działające identycznie jak ikony w oknie głównym oraz bardzo przydatną funkcję Get Volume Information podającą sporo danych dotyczących wybranego urządzenia. W menu Macro zgrupowane są opcje przeznaczone do kontroli współpracy programu z Arexxem pozwalające na stosowanie makrodefinicji. Na uwagę zasługuje też bardzo oryginalnie rozwiązana opcja About. Umieszczona w tym oknie animacja jest bardzo prosta, ale wywołuje spore wrażenie przy pierwszym kontakcie z programem. Nikt bowiem nie spodziewa się umieszczenia takiej



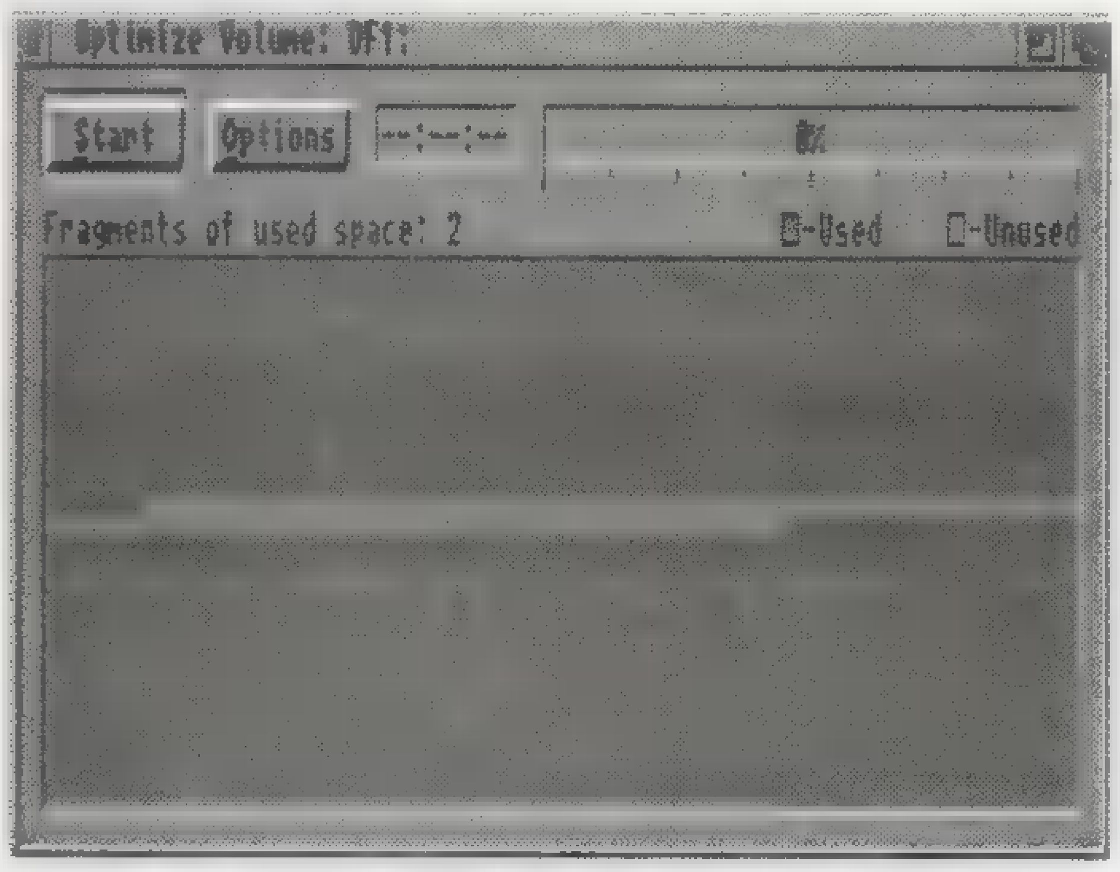
czysto rozrywkowej procedury, w bądź co bądź poważnym programie. Wszystkie opcje "robocze" umieszczone są w oknach pomocniczych wywoływanych za pomocą ikony lub z menu.

Blok Analyze and Repair Volume jest przeznaczony do naprawy struktury dysku. Spełnia więc te same funkcje co Analyst w AMI-BACK TOOLS i Repair oraz Validate w DISKSALV. Okno jakie pojawia się po jego uruchomieniu zawiera w dolnej części mały ekran informacyjny, na którym na bieżąco wyświetlane są informacje o wykonywanej operacji. Jest on bardzo wygodny w użyciu, gdyż jest wyposażony w "suwaki" pozwalające na przewijanie tekstu i dokładne przeanalizowanie wszystkich komunikatów, które pojawiły się podczas pracy. Powyżej znajduje się typowy "pasek" obrazujący zaawansowanie operacji. Dodatkowo parametr ten pokazywany jest liczbowo w procentach. W górnej części bloku umieszczono trzy gadżety. Start i Abort pozwalają na uruchomienie i zatrzymanie procesu, a kliknięcie na Options wywołuje dodatkowe okienko z parametrami. Można w nim ustawić opcje warunkujące przebieg procesu naprawy struktury, czyli zadeklarować sprawdzanie plików, bloków danych i co najważniejsze, włączyć lub wyłączyć naprawę dysku. Wbrew pozorom, wyłączenie naprawy dysku jest bardzo przydatne, gdyż pozwala na przeprowadzenie analizy dysku bez dokonywania na nim zapisu. Po wynikach takiej analizy można z dużym prawdopodobieństwem określić, czy operacja zakończy się sukcesem. Możliwości takiej brakowało w obu wcześniej opisywanych programach co zmuszało do wykonywania kopii bezpieczeństwa. Po uruchomieniu procesu trzeba uzbroić się w cierpliwość i czekać na zakończenie pracy programu. Oczekiwanie można sobie urozmaicić czytaniem komunikatów podawanych przez program. Jest ich dosyć dużo i użytkownik jest dokładnie informowany o aktualnie wykonywanej operacji.

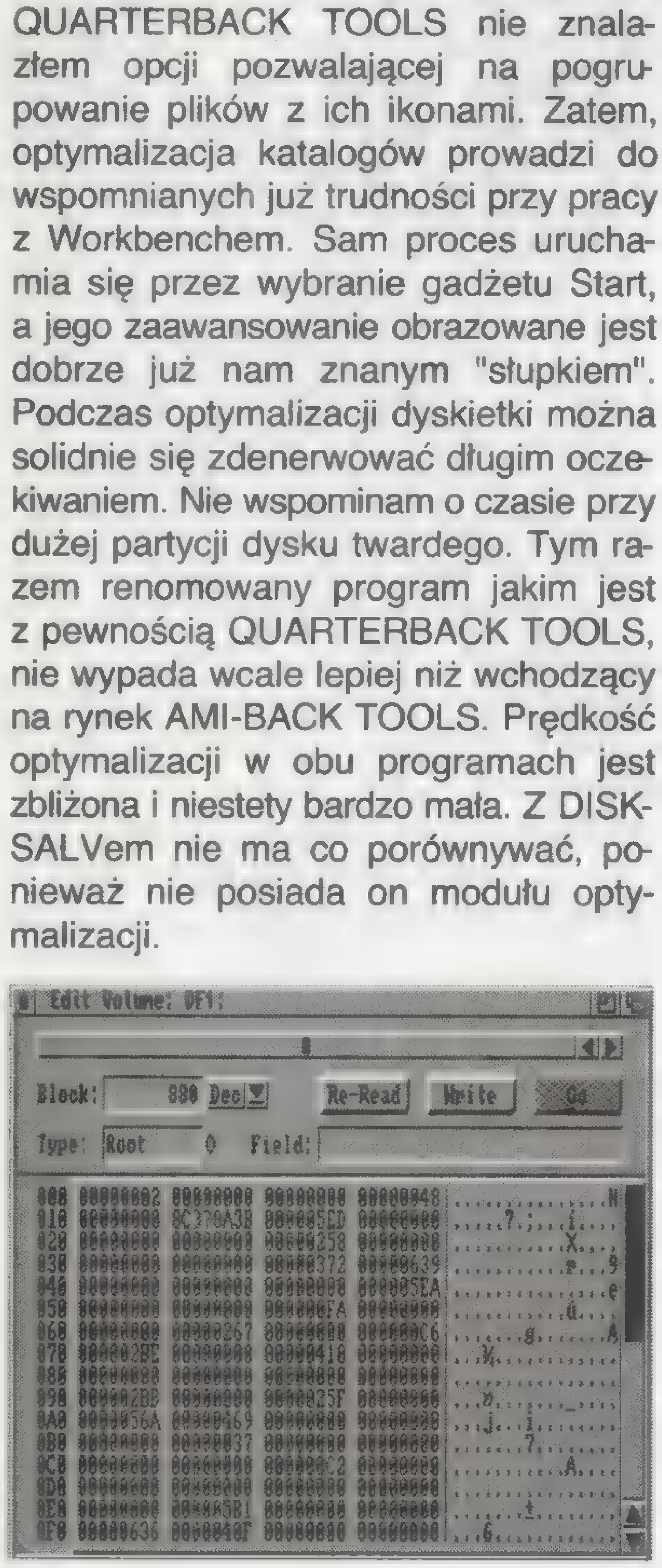
Kolejny blok to Recover Lost/Deleted Files. Przeznaczony jest on do odzyskiwania plików przypadkowo skasowanych i uszkodzonych. I tym razem można znaleźć podobne funkcje w obu poprzednio opisywanych programach. W AMI-BACK TOOLS jest to Reviver, a w DISKSALV Salvage i Undelete. Ok-



no jakie zobaczymy po przejściu do tego bloku jest zorganizowane podobnie jak w opcji poprzedniej. Nad oknem informacyjnym znajdziemy jednak więcej gadżetów. Najważniejszym jest "Recover to: ..." pozwalający ustalić czy odzyskane pliki będą nagrywane bezpośrednio na przeszukiwanym dysku, czy też zostaną przekopiowane na inne urządzenie. Zdecydowanie polecam to drugie rozwiązanie, gdyż jest znacznie bezpieczniejsze. Korzystając z gadżetu Options możemy podobnie jak poprzednio ustawić kilka parametrów dotyczących pracy programu, a kliknięcie na Start uruchomi proces przeszukiwania dysku. Po jego zakończeniu ujrzymy na ekranie spis plików odzyskanych lub wszystkich obecnych na dysku (zależnie od ustawienia w oknie Options). Standardowo do odzyskania program zaznacza wszystkie pliki i jeśli chcemy zrezygnować z "ratowania" jakiegoś trzeba go odznaczyć przez kliknięcie na jego nazwie lub użycie gadżetu Untag (po uprzednim wskazaniu jego nazwy). "Przycisk" Tag przeznaczony jest do zaznaczania plików, a Enter i Back umożliwiają poruszanie się po katalogach. Jeśli mamy zaznaczone właściwe pliki i katalogi, możemy kliknąć na Start i jeżeli jest ustawione nagrywanie na dysk przeszukiwany, pliki zostaną na nim utworzone. Jeżeli natomiast ustawiliśmy kopiowanie na inne urządzenie, to przed wybraniem Start trzeba jeszcze w odpowiednim polu wpisać właściwą ścieżkę dostępu. Z powyższego opisu wynika, że program QUARTERBACK TOOLS pozbawiony jest możliwości wykonania archiwizacji. Można jedynie odzyskane pliki przekopiować na inne urządzenie w standardowej postaci. Widać tu wyraźnie specjalizację i przerzucenie funkcji archiwizacji do oddzielnego programu, także wyspecjalizowanego, czyli QUARTERBACK.



Kolejny blok funkjonalny to Optimize Volume. Jest to optymlizer nieco podobny do dostępnego w AMI-BACK TOOLS. Większa część okna, pojawiającego się po jego uruchomieniu przeznaczona jest na graficzną prezentację mapy zajętości dysku. Gadżet Options, podobnie jak poprzednio wywołuje okienko z parametrami. Można ustawić kilka parametrów związanych z przebiegiem reorganizacji dysku (możliwości podobne do odpowiednich w AMI-BACK TOOLS). Niestety, i w QUARTERBACK TOOLS nie znalazłem opcji pozwalającej na pogrupowanie plików z ich ikonami. Zatem, optymalizacja katalogów prowadzi do wspomnianych już trudności przy pracy z Workbenchem. Sam proces uruchamia się przez wybranie gadżetu Start, a jego zaawansowanie obrazowane jest dobrze już nam znanym "słupkiem". Podczas optymalizacji dyskietki można solidnie się zdenerwować długim oczekiwaniem. Nie wspominam o czasie przy dużej partycji dysku twardego. Tym razem renomowany program jakim jest z pewnością QUARTERBACK TOOLS, nie wypada wcale lepiej niż wchodzący na rynek AMI-BACK TOOLS. Prędkość optymalizacji w obu programach jest zbliżona i niestety bardzo mała. Z DISKSALVem nie ma co porównywać, ponieważ nie posiada on modułu optymalizacji.



Ostatnim blokiem jest Edit Volume, czyli edytor dysku. Znaczącą część okna jakie pojawia się po jego uruchomieniu zajmuje prezentacja danych zawartych na dysku. Zrealizowana jest ona w sposób raczej standardowy. Z lewej strony widzimy dane w trybie hexadecymalnym, a przy prawej krawędzi to samo w kodach ASCII. Do dyspozycji mamy także suwak pozwalający na stosunkowo małej powierzchni okna obejrzeć zawartość całego bloku. W połączeniu z możliwością bezpośredniego dokonywania zmian w danych i to zarówno reprezentowanych szesnastkowo jak i kodami ASCII, daje to dużą swobodę w pracy. Dokonane zmiany można oczywiście utrwalić za pomocą gadżetu Write. Nie jest to jednak wszystko co oferuje użytkownikowi ten edytor. Przy górnej krawędzi okna mamy jeszcze jeden suwak przeznaczony do wybierania potrzebnego bloku dysku, którego numer pokazywany jest na ekranie w stosownym polu. Wraz z pojawieniem się okna edytora uzyskujemy także dostęp do nowego menu "z belki". Pod nazwą Edit znaleźć można kilka opcji dobrze znanych wszystkim korzystającym z programów do edycji dysku. Nadmienię tylko, że jest możliwość obliczenia nowej sumy kontrolnej, poszukiwania zadeklarowanego ciągu znaków, skoku do wybranego bloku, itp. Edytor zawarty w QUARTERBACK TOOLS oceniam naprawdę wysoko i uważam, że pod pewnymi względami przewyższa on niektóre programy specjalizowane, przeznaczone wyłącznie do dokonywania zmian bezpośrednio w danych zawartych na dysku.

Podsumowując opis chciałbym zaznaczyć, że QUARTERBACK TOOLS to naprawdę starannie napisany program. Zadbano o wszystkie ważne szczegóły z estetyczną szatą graficzną włącznie. Na zakończenie powinienem wskazać program najlepszy z pośród trzech tu opisywanych. Dokonanie takiego wyboru nie jest jednak możliwe, ponieważ każdy z nich ma swoje wady i zalety. Z pewnością AMI-BACK TOOLS jest najbardziej uniwersalny. Oferuje szeroki zakres operacji, od odzyskiwania uszkodzonych plików, przez optymalizację dysków, po całkiem przyzwoity archiwizer, a wszystko to zgrupowane w jednym miejscu i dostępne po jednym kliknięciu myszką. QUARTERBACK TOOLS natomiast imponuje swoją renomą i znaną jakością wykonywanej pracy. Działa raczej wolno, ale za to bardzo dokładnie i skutecznie. Jest przy tym dopracowany w najdrobniejszych szczegółach, co dodatkowo upewnia użytkownika o jego wysokiej jakości. Oferuje ponadto edytor dysku, którego nie ma w pozostałych programach. Mimio braku (w chwili obecnej) modułu archiwizacji QUARTERBACK TOOLS to potężne narzędzie mogące zadowolić nawet najbardziej wybrednych. Pojawia się tylko problem ceny jaką trzeba zapłacić za oba wyżej wspomniane programy. I pod tym względem najlepiej wypada DISKSALV. (program Shareware, dostępny za symboliczną opłatą). Możliwości jakie przy tym oferuje są całkiem spore. Z AMI-BACK TOOLS przegrywa brakiem optymlizera i bezpośredniej archiwizacji na dyskietki. W stosunku do QUARTERBACK TOOLS brakuje mu optymlizera i edytora dysku. Lecz jak na program Shareware swoje podstawowe zadanie, czyli odzyskiwanie uszkodzonych plików, spełnia doskonale. Może mieć to spore znaczenie w przyszłości, zapowiadanej jako era tępienia piratów komputerowych. Przy zakupie oryginalnych programów wybór może paść właśnie na DISKSALVa ze względu na cenę. Jak widać podjęcie decyzji jest raczej trudne i powinno być uzależnione od potrzeb danego użytkownika oraz zasobności jego kieszeni.

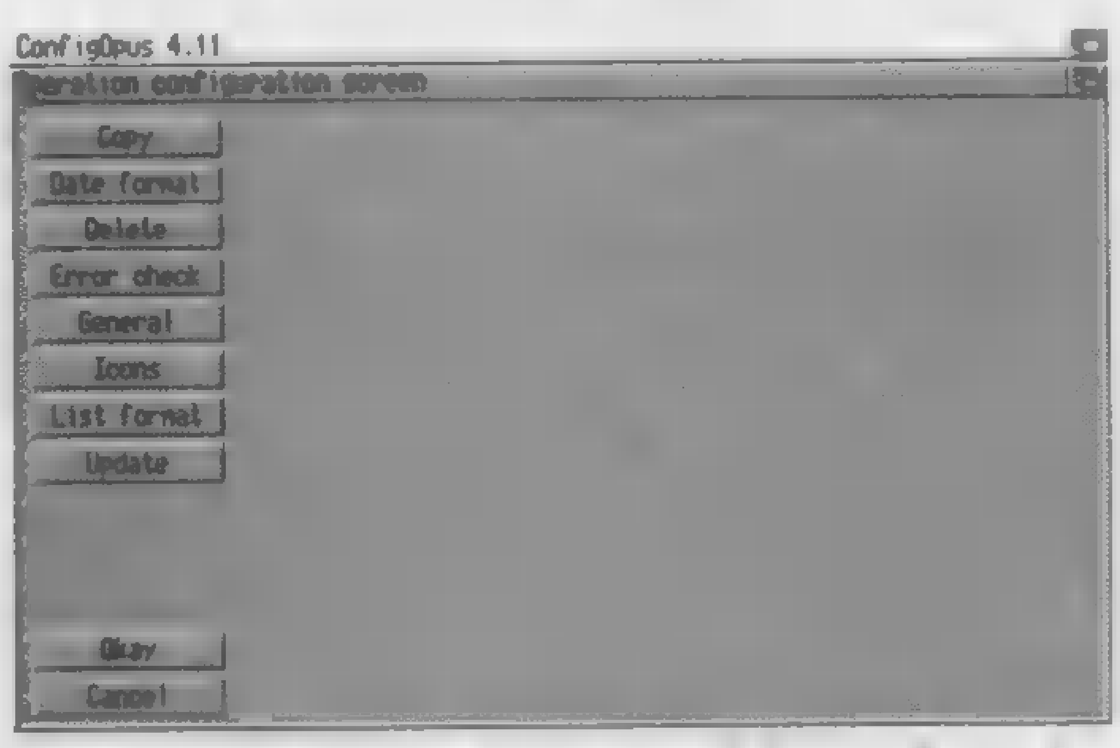
Na tym kończę artykuł i mam nadzieję, że zawarte w nim informacje pozwolą wielu czytelnikom AMIGOWCA na wybranie programu odpowiedniego do swoich potrzeb. Chciałbym też życzyć wszystkim, aby jak najrzadziej musieli korzystać z opisywanych programów. Utrata pieczołowicie zbieranych danych lub efektu wielogodzinnej pracy nigdy nie jest przyjemna. A jeśli już takie nieszczęście się wydarzy, to warto przypomnieć sobie o tych trzech programach, które mogą wybawić nas z opresji. Do opisu zostały wykorzystane: AMI-BACK TOOLS V1.04 firmy Moonlighter Software Development, Inc, DISKSALV Version 2, którego autorem jest Dave Haynie oraz QUARTERBACK TOOLS V2.0 firmy Central Coast Software. Wszystkie z 1993 r. ❑

OPUS v4.11 cz.2

Tomasz Kokoszczynski

W poprzednim miesiącu przerobiliśmy początki konfigurowania nowej wersji Opusa 4.11. A wiele on się zmienił, oj wiele. Największe zmiany zaszły chyba przy współpracy programu z systemem, a tym zajmiemy się właśnie dzisiaj. Wróćmy do panelu konfiguracyjnego.

Operacje czas zacząć

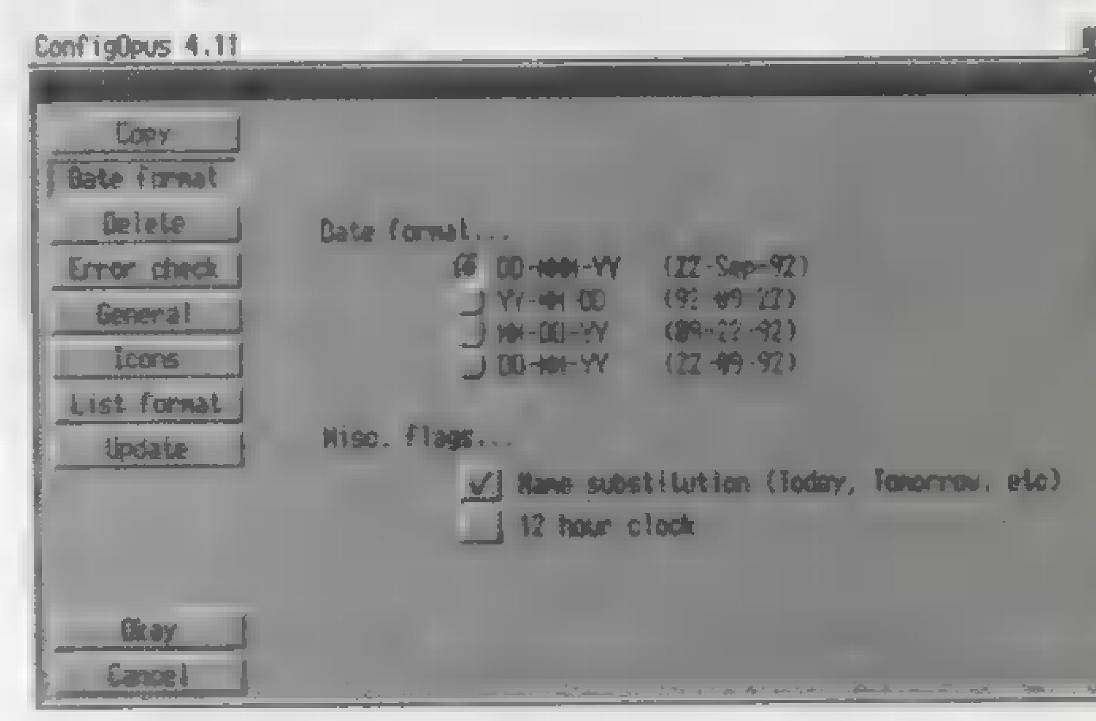


Z panelu operacyjnego wybiera się tryby przeprowadzania operacji.

Panel operacyjny służy do definiowania podstawowych operacji Opusa. Należą do nich oczywiście: kopiowanie i kasowanie. Oprócz tego możemy dokonać kilku ustaleń tego, co pojawi się w oknie programu i w jakiej kolejności będą porządkowane wyświetlane pliki. Na początek przejdźmy do najprzyjemniejszej funkcji Opusa jaką jest kopiowanie. Jej konfiguracja następuje oczywiście poprzez wciśnięcie klawisza Copy. Musicie wiedzieć, że tak naprawdę kopiowanie wcale nie jest tak proste jak jedzenie bułki z masłem. Aby skopiować plik(i) czasami niezbędna jest informacja, czy zmieszczą się one na docelowym nośniku. Możemy nakazać programowi sprawdzenie tego przed każdym kopiowaniem. Jeśli plik się nie zmieści program poinformuje nas odpowiednim komunikatem. Nie przeszkadza to jednak, abyśmy próbowali go gdzieś wcisnąć (np. na miejsce pliku starego o tej samej nazwie). Czasami przy kopiowaniu istotne jest zaznaczenie, które pliki zostały już skopiowane. W takim przypadku nakazujemy Opusowi wstawienie bitu archiwizacji (A). Oprócz kopiowania właściwego pliku możemy dodatkowo skopiować kilka innych informacji. Czasami warto kopiować pliki z oryginalną datą, czasami zaś warto wprowadzić do nich czas, w którym dokonujemy kopiowania. Jest to szczególnie ważne przy porządkowaniu większych ilości plików, a Opus umożliwia nam przecież przeglądanie plików ustawionych datami (o czym za chwilę). Oprócz daty każdy plik zawiera informacje dotyczące bitów protekcji oraz komentarz. Bity protekcji zabezpieczają pliki przed przypadkowym skasowaniem. Natomiast komentarze polecam gorąco wszystkim posiadaczom twardych dysków. Czasami okazuje się, że mamy np. kilkadziesiąt rodzajów różnych czcionek i nie wiemy, które możnaby skasować. A wystarczyłoby tylko zaopatrzyć każdorazowo kopiowane czcionki w nazwę programu, do którego należą. Do tego właśnie idealnie nadaje się miejsce stworzone przez komentarz.

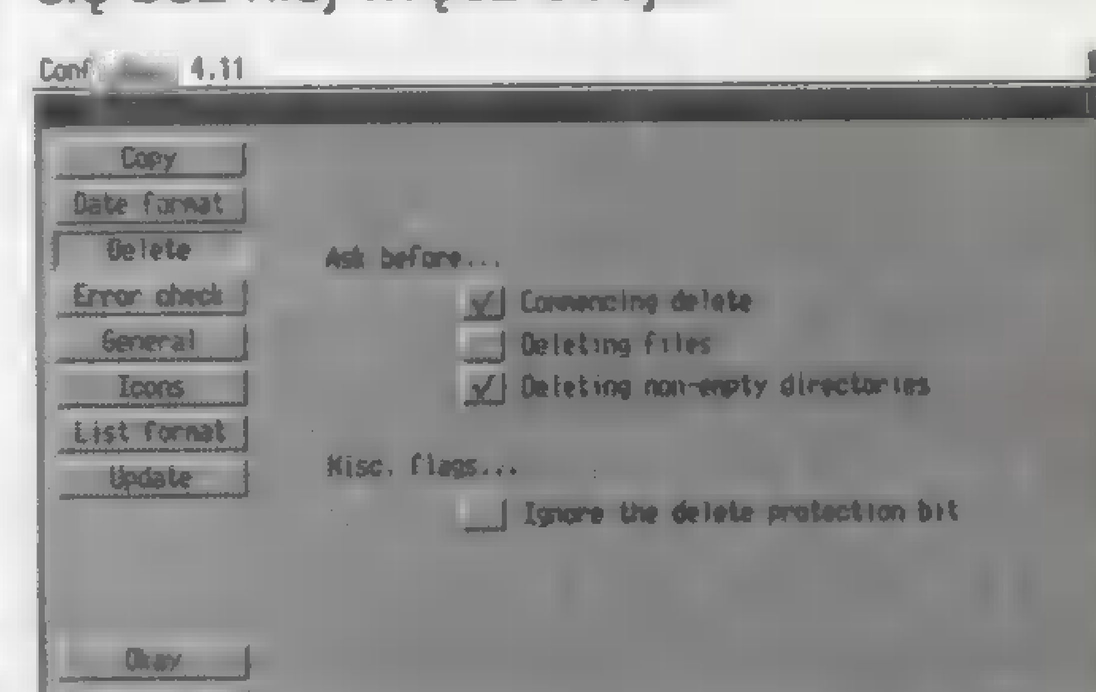
Oprócz kopiowanych informacji powinniśmy ustalić zachowanie się programu w momencie, gdy będzie kopiował plik z jednoczesnym kasowaniem pliku już istniejącego o tej samej nazwie. W tym samym katalogu nie mogą bowiem istnieć dwa pliki o identycznej nazwie i w przypadku zwykłego użycia polecenia "copy" możemy potem nieprzyjemnie się zdziwić. Chyba najrozsądniejszym wyjściem jest wybranie opcji "Ask before replacing" (pytaj przed kasowaniem). Zdawanie się na losowe kasowanie plików starszych może się okazać czasami zgubne. Nie widzę też powodu, żeby całkowicie wykluczyć nadpisywanie jednych plików na miejsce drugich. Opcje te wydają mi się troszeczkę nadmiarowe. No, ale co wolność, to wolność.

Nie powinniście mieć większych problemów z ustawieniem odpowiedniego trybu wyświetlania daty (Date format).



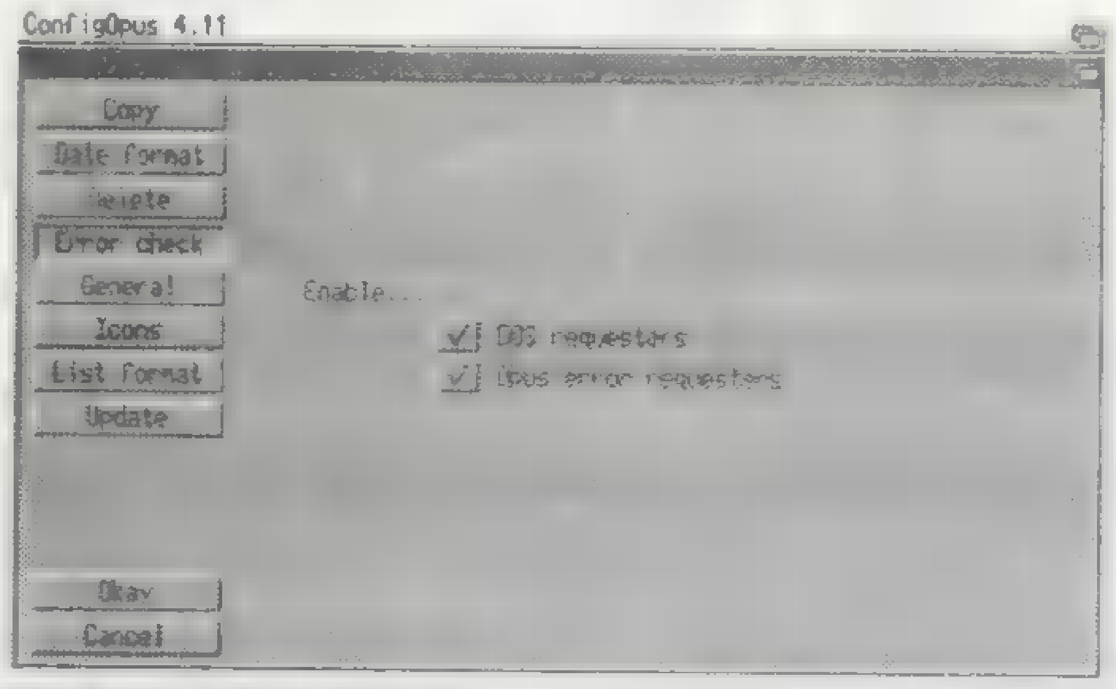
Gorąco polecam wstawienie opcji podstawiania zamiast daty nazw: Today (dzisiaj), Tomorrow (jutro) i Yesterday (wczoraj). Czasami po prostu łatwiej można zorientować się w istniejących plikach, gdy zamiast cyferek mamy informację słowną.

Kasowanie plików jest smutną koniecznością, ale czasami nie można się bez niej wręcz obejść.



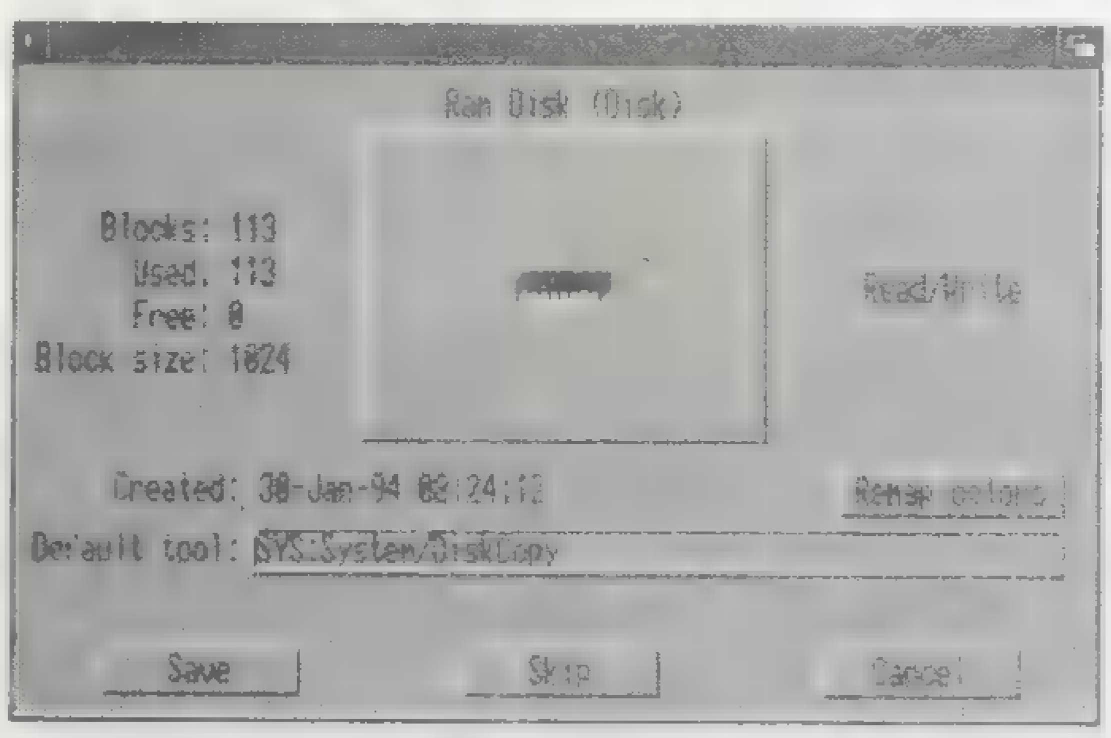
W opcji Delete możemy nakazać programowi, kiedy ma nas pytać, czy skasować dany plik. Może to mieć miejsce przed całym procesem, przy każdym kasowanym pliku lub wreszcie przy kasowaniu zapełnionych szuflad. Szczególnie ta ostatnia opcja może niejednokrotnie uchronić zawartość Waszych zbiorów, które być może

zdarzy się Wam poddać pomyłkowej dekapitacji.



Ważne informacje.

Jeśli chodzi o następną pozycję, pozwala ona ustalić sposób informowania nas o różnego rodzaju "wypaczeniach systemowych". Radzę odhaczyć obie opcje, a wtedy program wybierze najwłaściwyszy sposób, aby nas o czymś poinformować.



Co robić gdy...

Odpowiednie znaczki wstawione w opcji General informują program, co ma zrobić gdy:

- wciśniemy myszkę raz nad plikiem i szybko ją przemieścimy do drugiego okna, gdzie klikniemy ponownie (Click-M-Click)

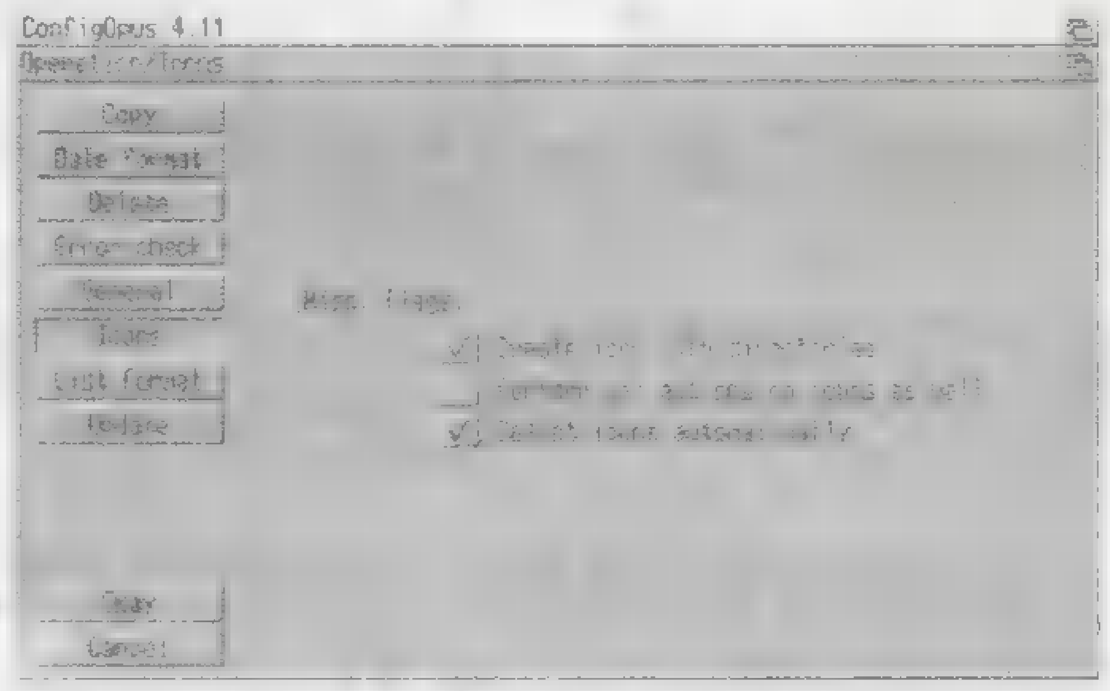
- istnieje .info do pliku (ma je wtedy po prostu wyświetlić - Display info).

- wciśniemy dwa razy lewy klawisz (LMB) nad jakimś plikiem (File double-click)

- wciśniemy prawy klawisz (RMB) - może wtedy zdarzyć się, że jeżeli będziemy przy brzegu okna, jego zawartość przesunie się.

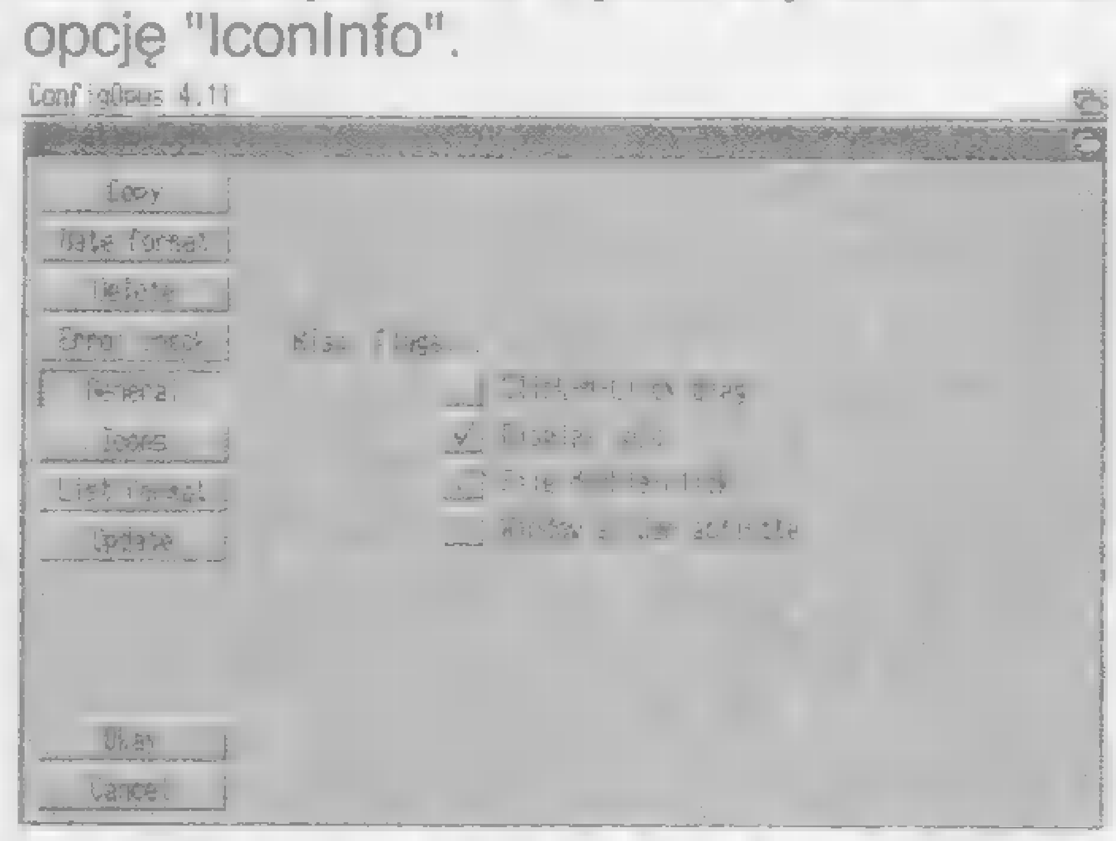
Opcje te są bardzo istotne dla wygodnej pracy z programem. Dzięki nim możemy ustalić swoje własne środowisko pracy. Przecież łatwiej jest przeczytać dany tekst, czy odegrać sobie modulik z SoundTakiegośTam klikając dwa razy na odpowiedni plik, niż zaznaczyć ten plik, zjechać myszą do dołu i wybrać to co ma się akurat z nim zdarzyć. Zdarza się, że gramy tekst i wczytujemy muzyczkę, a przecież program przeważnie jest w stanie zdefiniować plik i wybrać dla niego odpowiednie działanie. Wyświetlanie .info jest sprawą dyskusyjną. Gdy mamy tego faktycznie za dużo, możemy sobie dać spokój. Osobiście lubię widzieć wszystkie pliki i dlatego mam tę opcję włączoną.

Przesuwanie zawartości okna przy pomocy RMB może czasami okazać się zbyt uciążliwe. Już nieraz zdarzyło mi się, że był to niezamierzony efekt mojej nieuwagi, a nie celowe działanie.



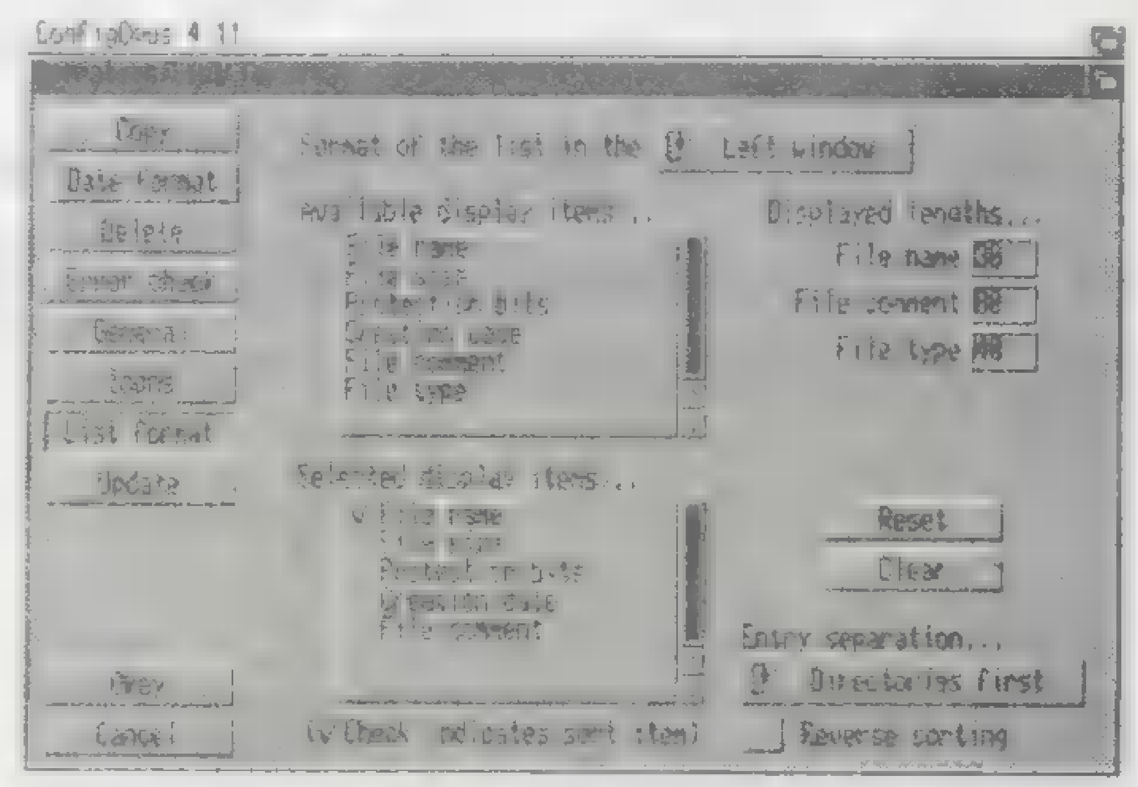
Z ikonkami trzeba dobrze żyć.

Opcja Icons umożliwia wybór działań na ikonach. Jak zapewne wszyscy wiecie ikony są kreowane sztucznie (po wybraniu odpowiedniej opcji w systemach od 2.0 wzwyż) lub jeśli istnieje do nich dodatkowy plik z nazwą, jak nazwa pliku i dopiskiem ".info". Plik taki zawiera rysunek ikony i podstawowe informacje o pliku. Wszystkim poważnym użytkownikom amigowego systemu polecam stosowanie ikon. Jest to wygodne i szybkie, a poza tym ładnie wygląda. Opus, mimo że sam działa na nazwach, może także wykonywać odpowiednie operacje na ikonach. Może je tworzyć wraz z tworzonym plikiem, a ich wygląd może być zróżnicowany w zależności od zawartości pliku. Program posiada automatyczny schemat przydzielania różnych typów ikon do odpowiednich programów. I tak inaczej będzie wyglądała ikona programu, a zupełnie inaczej ikona pliku spakowanego "lharkiem". Oczywiście, jak to w demokracji bywa, istnieje pełna możliwość wyboru własnych, czasami głupszych, czasami mądrzejszych rysunków, czyli sporo uciechy przy takich sobie zwykłych czynnościach. Rodzaje ikon wybieramy z opcji System z podpunktu "Icons". Opus ma możliwość dokonywania zmian informacji zawartej w ikonach. W tym celu wystarczy uruchomić opcję "IconInfo".



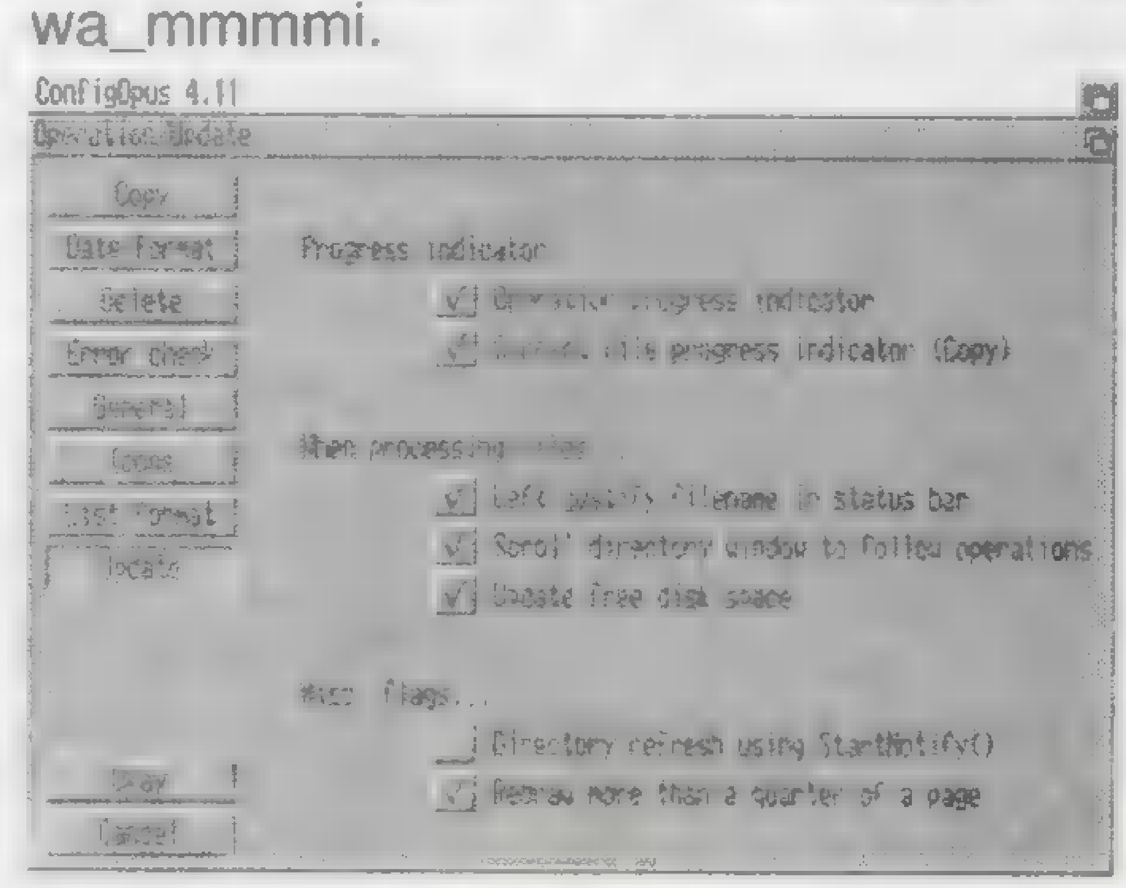
Tak zmieniamy to, co kiedyś można było zmienić tylko w Workbenchu.

Istnieje w Opusie opcja, której brakuje mi przy wczytywaniu czy zgrywaniu różnych plików z innych programów. Opcja ta należy do punktu Operation i nosi nazwę List Format. Dzięki niej możemy ustalić w jaki sposób mają być wyświetlane pliki. Czy będą uporządkowane alfabetycznie, czy też datami bądź wielkościami. Pragnę Wam zwrócić uwagę, że aby nastąpiła zmiana w wyświetlaniu zawartości jakiegoś katalogu, należy ponownie go wczytać (RMB nad gadżetem R). Poza tym możemy ustalić co ma być wyświetlane i ile miejsca czemu się należy. Przy możliwości przesuwania środkowej listwy pomiędzy oknami, możemy dograć sobie wszystko.



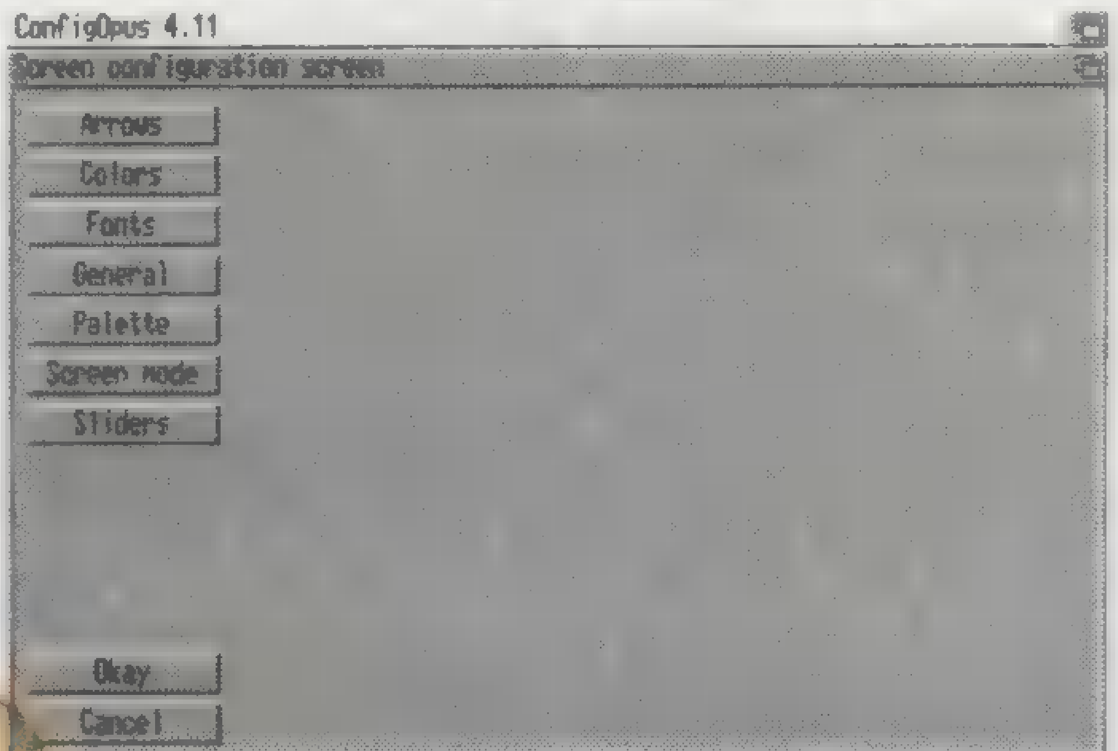
Opusiku powiedz przecie, jaki plik jest największym ... na moim dysku.

Odświeżanie wszystkiego co tylko możliwe jest jedną z życiowych dewiz twórców Opusa. Odświeżają nie tylko i tak już coraz (za)doskonalsze wersje programu, ale sam program może odświeżać na różne sposoby wszystko co do niego wczytano. W opcji Update można bez obawy włączyć wszystko. Ciekawa jest opcja, której brakowało w poprzednich wersjach - wskaźnik postępów przy kopiowaniu danego pliku. Właściwie informacja ta niczego nie zmienia, ale ładnie się ogląda jak się kopiuje jakiś wielki plik, a jeszcze ładniej jak się kopiują małe pliczki. Opócz tego możemy czasami stwierdzić, że coś jeszcze nam pracuje w systemie, bo to co kiedyś kopiowało się ułamek sekundy, teraz kopiuje się tak straaaasznie jakoś wolno i zrryyy_wawa_mmmmi.



Najważniejsza jest ręka na pulsie.

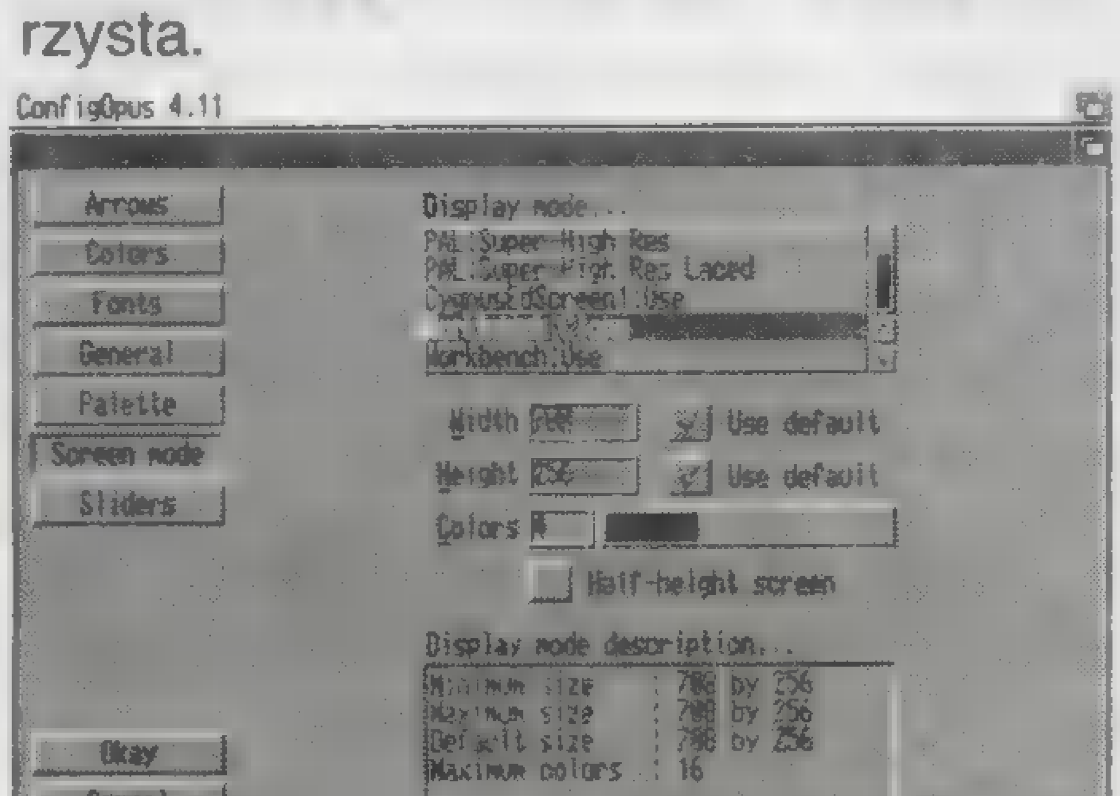
Ekranizacji ciąg główny.



Ekranizacja.

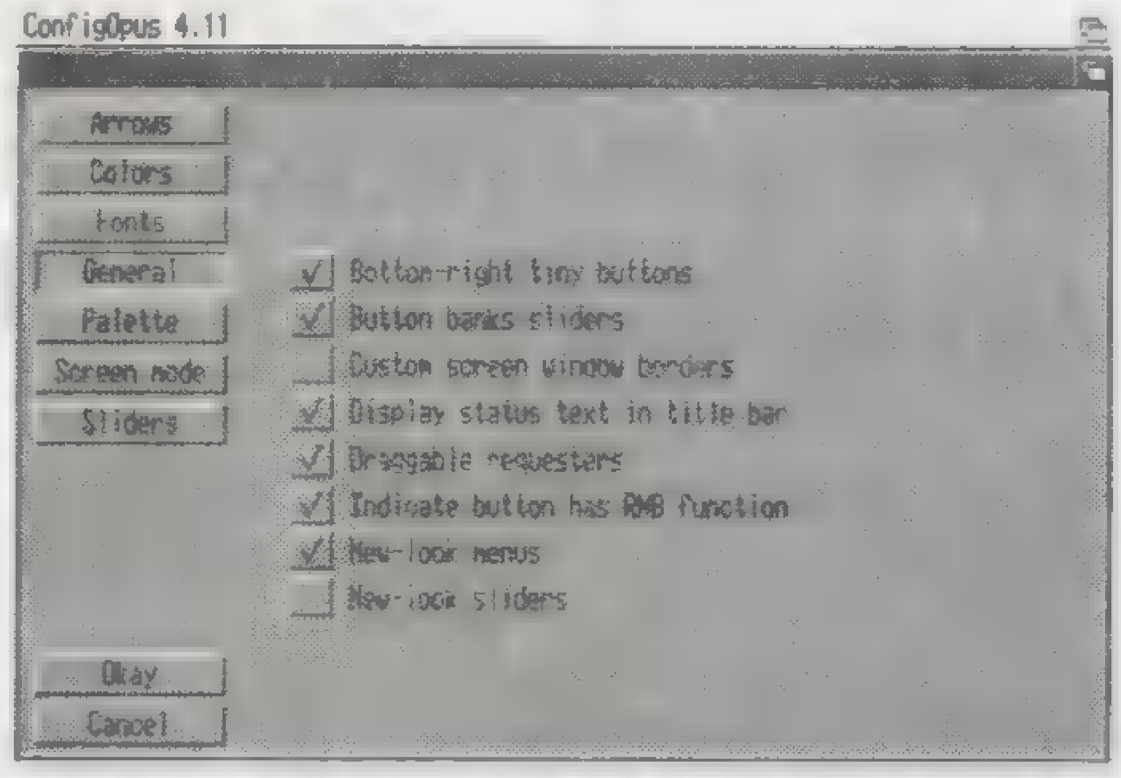
Ekranizacja to oczywiście wszystko co się wiąże z ekranem Opusa. Mamy do niej dostęp poprzez opcję Screen. Dzięki niej można realizować prawie wszystkie szalone pomysły. Oczywiście w dalszym ciągu nie ma możliwości rysowania aniołków, ale da się ustalić wygląd strzałek, ilość kolorów, no i oczywiście wielkość ekranu Opusa. Szczególnie wygodna jest, dla wszystkich pracujących w niestandardowej rozdzielczości programów, możliwość skorzystania z już istniejącej rozdzielczości np. Workbencha. W ten prosty sposób ekran Opusa będzie taki sam jak ekran Workbencha. Opcja ta nie została niestety uskuteczniona w panelu konfiguracyjnym (działa zawsze w tej samej rozdzielczości) i do dyspozycji mamy tylko standardową rozdzielczość. No, ale to akurat można wybaczyć twórcom Opusa. Muszę powiedzieć, że byłby to całkiem frapujący pomysł: program konfiguracyjny dla programu konfiguracyjnego...

Za to samego Opusa można zmniejszyć nawet o połowę tak, że będziemy mieli odsłoniętą drugą część ekranu. W zaufaniu powiem Wam, że nie jest to wcale wygodne i mało kto z tego korzysta.



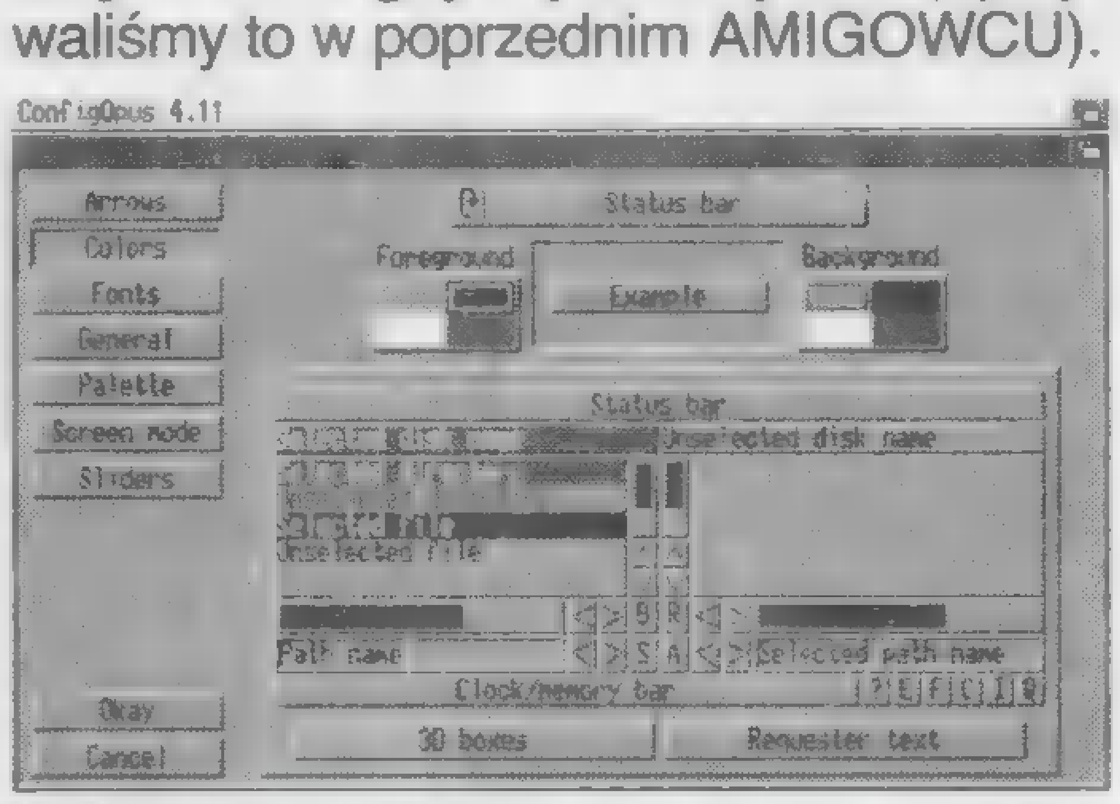
Wybór wielkości ekranu staje się tak prosty.

Dosyć istotne w panelu konfigurującym ekran są oczywiście opcje ogólne. Umożliwiają one ustalenie, czy dany tekst będzie wyświetlany. Mówiąc "dany tekst" rozumiemy np. zegar, drobne klawisze w prawym dolnym rogu służące do wywoływania różnych opcji, itp. Ciekawym udogodnieniem jest wskazywanie, czy dany przycisk ma drugą, "ukrytą" funkcję (Indicate button has RMB function) - klawisze, które mają taką funkcję zostaną zaopatrzone w zagięty rożek.



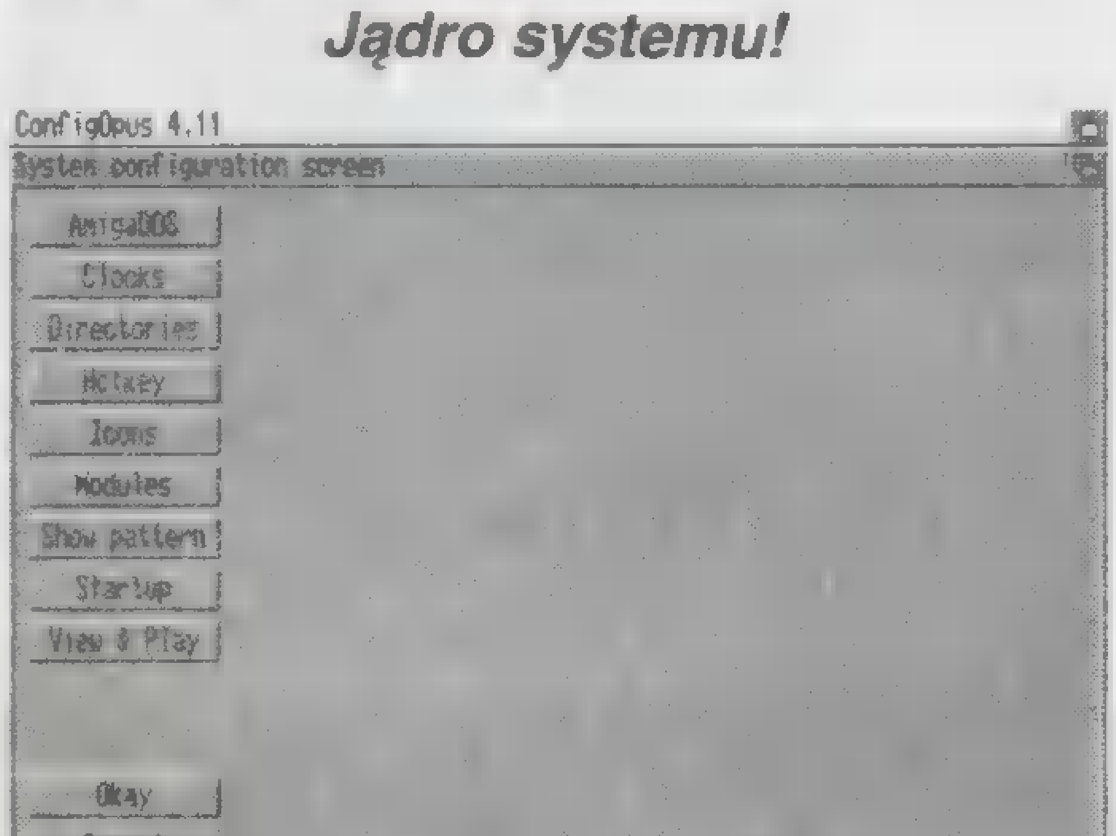
Ogólnie też warto niektóre rzeczy wiedzieć.

Oczywiście jak we wszystkich poprzednich wersjach istnieje możliwość ustawienia kolorów Opusa. Możemy wybrać od 4 do 16 kolorów. Liczbę kolorów określamy w opcji Screen Mode. Gdy już to ustalimy, (ale nie przesadzajmy, gdyż 16 kolorów to naprawdę nieco za wiele), możemy dobrać rodzaje kolorów (duża paleta, oj duża). Mi osobiście odpowiada w zupełności ustawienie standardowe. Kolory możemy ustawić dla prawie wszystkich elementów takich jak: wygląd pudełek, przycisków, jak mają być podświetlane pliki, itd. Oczywiście osobno podczas definiowania własnych klawiszy ustalamy jaki mają mieć zestaw kolorów, gdy są aktywne lub gdy są nieaktywne (opisywaliśmy to w poprzednim AMIGOWCU).



Kolorowe na zielonym (monitorze) i tak nie wyjdzie.

Jądro systemu!



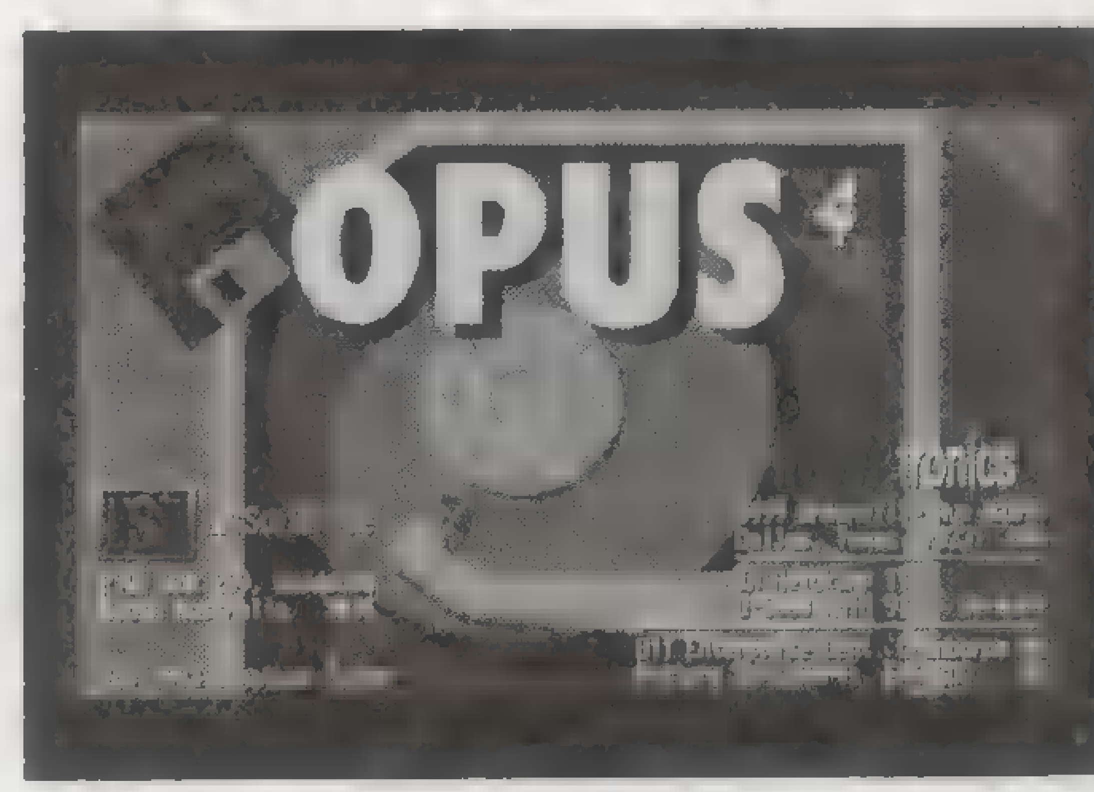
No i dotarliśmy do jądra.

Jądro Opusa leży nie w samym systemie jego działania, ale właśnie współdziałania. Z systemem Amigi oczywiście. Wiecie dlaczego nie ma Opusa na komputery innych typów? No właśnie dlatego. A propos Amigi to firma produkująca i sprzedająca Opusa ma bardzo ciekawą kampanię reklamową. Oprócz tego, że jej przedstawiciele pojawiają się na wszystkich targach i są bardzo mili dla klientów, firma wypuściła znaczki do przyczepienia (broszki) z napisem:

"Jay Miner uses directory OPUS... and so do I!".

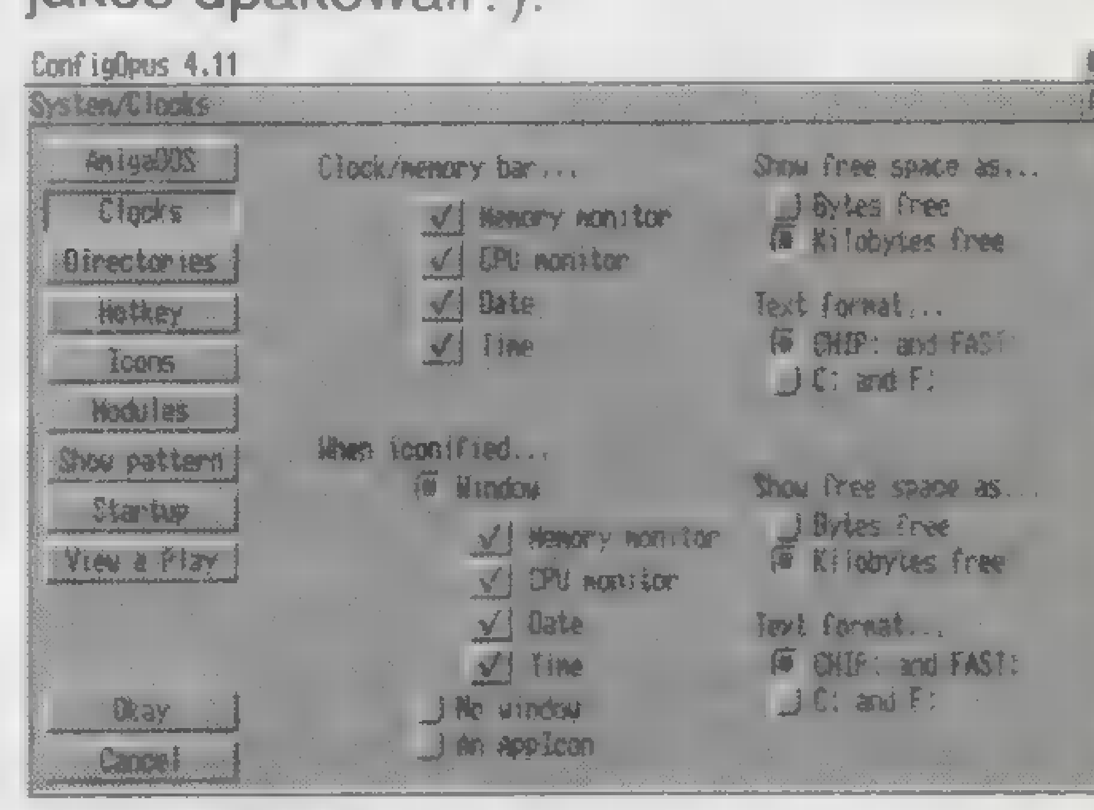
Zaś wszyscy zapaleni użytkownicy Amigi wiedzą, że Jay Miner jest twórcą tego komputera i nawet przez głowę by mu nie przyszło, żeby używać programów gorszych, niż te które są najlepsze. W tym i Opusa. Ciekawe, czy Jay zgodził się na tę kampanię przez sympatię do firmy, czy też raczej tkwią za tym jakieś pieniądze. Ja stawiam na sympatię.

Ciekawy jest też sposób wyświetlania informacji o programie. Po wybraniu opcji About pojawia się:



Coś takiego...

Wypada mieć tylko nadzieję, że rysunek ten nie jest za każdym razem ładowany do pamięci (z moich obserwacji wynika, że jednak jest - może go jakoś upakowali?).



Czas to też Opus.

Wróćmy do systemu. Na początku definiujemy okienko jakie będzie za każdym razem otwierane, gdy uruchomimy z Opusa jakiś program. Niestety programy uruchamiając się blokują Opusa, co czasami jest nam bardzo nie w smak. Oczywiście można je urucho

mić za pomocą polecenia RUN, które uwolni Opusa, ale o ileż przyjemniejsze jest uruchamianie za pomocą klikania myszką! Opus pozwala na prawie całkowicie dowolne podefiniowanie wyświetlania stanu najaktualniejszych informacji.

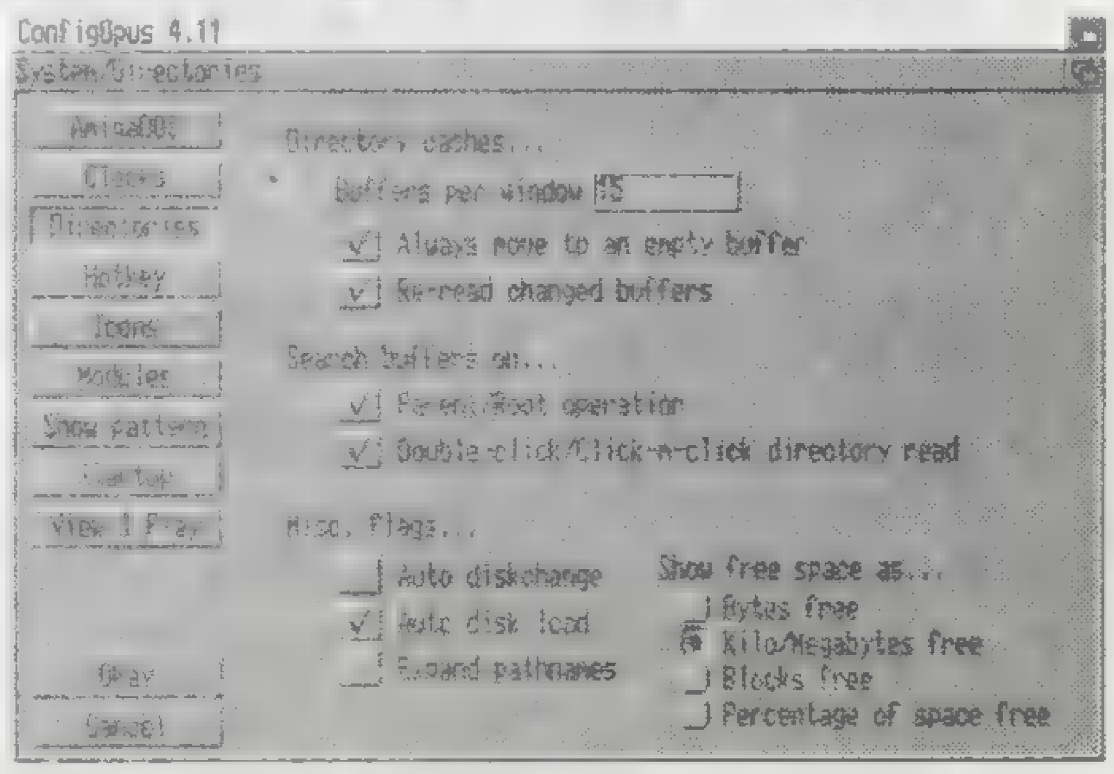
Możemy ustalić w jaki sposób będzie podawana wielkość wolnej pamięci. Zarówno w czasie pełnego funkcjonowania programu (forma rozwinięta), jak i w czasie jego ikonifikacji (forma zwinięta). Pragnę tu nadmienić, że Opus uaktualnia swoje informacje niezwykle szybko - o wiele szybciej niż sam system. Oczywiście możemy zrezygnować z ikony w stanie zikonifikowania (wywołanie tylko przez klawisze lub programik Exchange) lub, aby pokazywała się nam prawdziwa ikona, która pozwoli na przywrócenie aktywności programu. Wszystko jest fajnie, gdy używamy listwy programu. Jej pozycja zostaje zapamiętana tak, że gdy ją raz ustawimy na właściwym miejscu to już tam automatycznie pozostanie (będzie pojawiać się w przypadku ponownej ikonifikacji). Natomiast nie przewidziano możliwości zapamiętania pozycji ikony. Tradycjonalistom odradzam jej używanie, gdyż szybko wpadną w złość.



Dosyć ciekawa jest funkcja monitorująca pracę procesora (CPU monitor). Swego czasu był na Amigę programik rozprowadzany na dysku Extras, który wabił się PM, a jego ikonka była w jakiś sposób związana z króliczkiem.

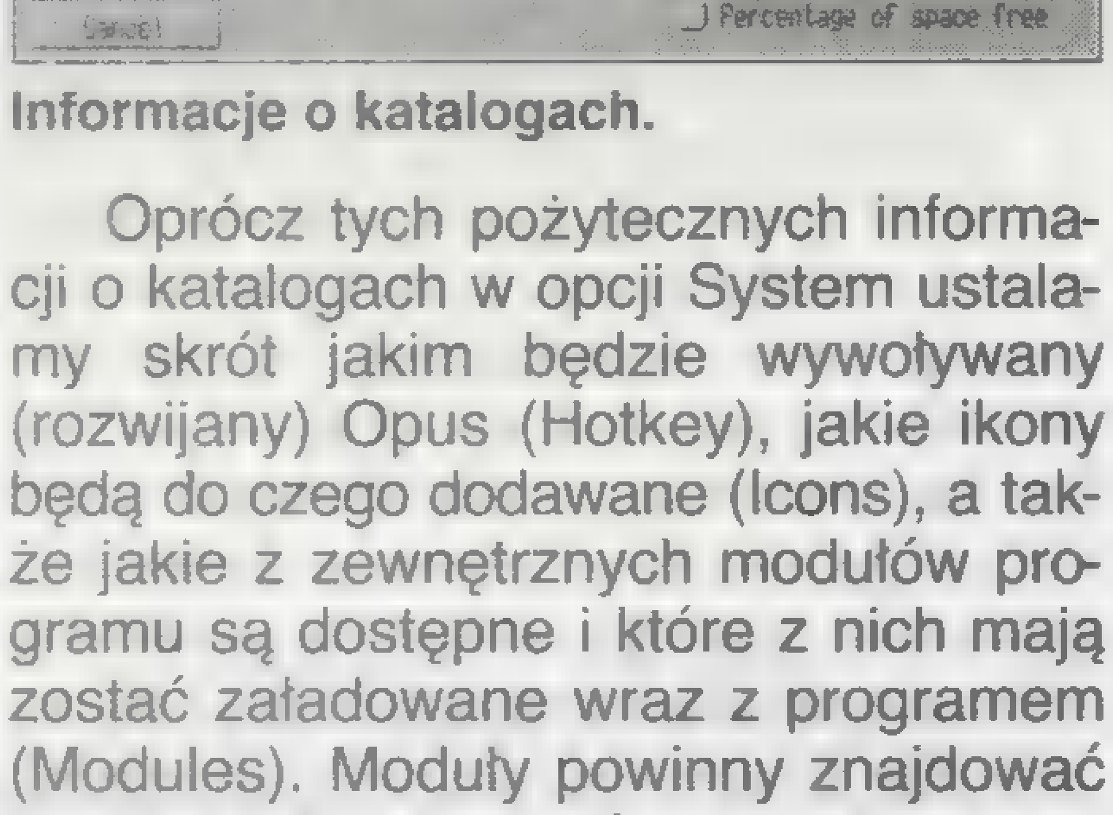
Program ten rysował wykres "EKG" głównego procesora komputera. Funkcja ta spełnia podobną rolę, tyle że stan zajęcia procesora pokazuje w procentach. Może to okazać się szczególnie ważne dla tych, którzy używają większej ilości programów na raz. Oczywiście programy stosujące masę obliczeń jak np. Imagine, natychmiast zajmą całą możliwą "przestrzeń" roboczą. Istnieją jednak takie programy, które będą pracowały na pół gwizdka i wtedy zaraz się o tym dowiemy.

Jedną z najważniejszych decyzji przy konfiguracji Opusa jest ustalenie jak program ma się obchodzić z danymi o ścieżkach. Ścieżki to oczywiście opisy drogi jaką musimy przebyć przez wszystkie katalogi, aby dotrzeć do katalogu bądź pliku żądanego. Możemy nakazać programowi pamiętanie np. 15 buforów. Oczywiście zajmie to całą masę pamięci, ale dostęp do tych danych będzie natychmiastowy. Opcja ta jest wroga wszelkim wolniejszym nośnikom i osoby posiadające jedną stację mogą ją przeklinać. Wydaje mi się, że uaktywnienie opcji Auto disk load może być całkiem dobrym antidotum na ten problem. Opcja ta spowoduje, że po zmianie dysku w stacji program będzie nas pytał, czy chcemy zaktualizować dane, czy też nie. Nawet posiadacze dwóch stacji "jakoś" spamiętają czy wymienili dyskietkę, czy też mają w stacji ciągle stary dysk. Przy zapamiętywaniu większej ilości danych polecam włączyć aktualizację zmienionych buforów, gdyż w przeciwnym wypadku zrobi się taki bałagan, że szkoda gadać. Opcja ta uchroni nas także przed wyświetlaniem stanu katalogu sprzed pięciu kopiowań.



Informacje o katalogach.

Oprócz tych pożytecznych informacji o katalogach w opcji System ustalamy skrót jakim będzie wywoływany (rozwijany) Opus (Hotkey), jakie ikony będą do czego dodawane (Icons), a także jakie z zewnętrznych modułów programu są dostępne i które z nich mają zostać załadowane wraz z programem (Modules). Moduły powinny znajdować się albo w katalogu głównym, albo w katalogu Modules. Są one czasami bardzo przydatne, gdyż można uruchomić je poza Opusem, jako oddzielne taski. Jest to niesamowicie wygodne, ponieważ nie blokujemy samego programu. Z modułów można korzystać jak z normalnych programów. Mała uwaga: jeśli program konfiguracyjny zostanie uruchomiony z Workbencha, to oczywiście po wyjściu z niego nie powrócimy do Opusa.

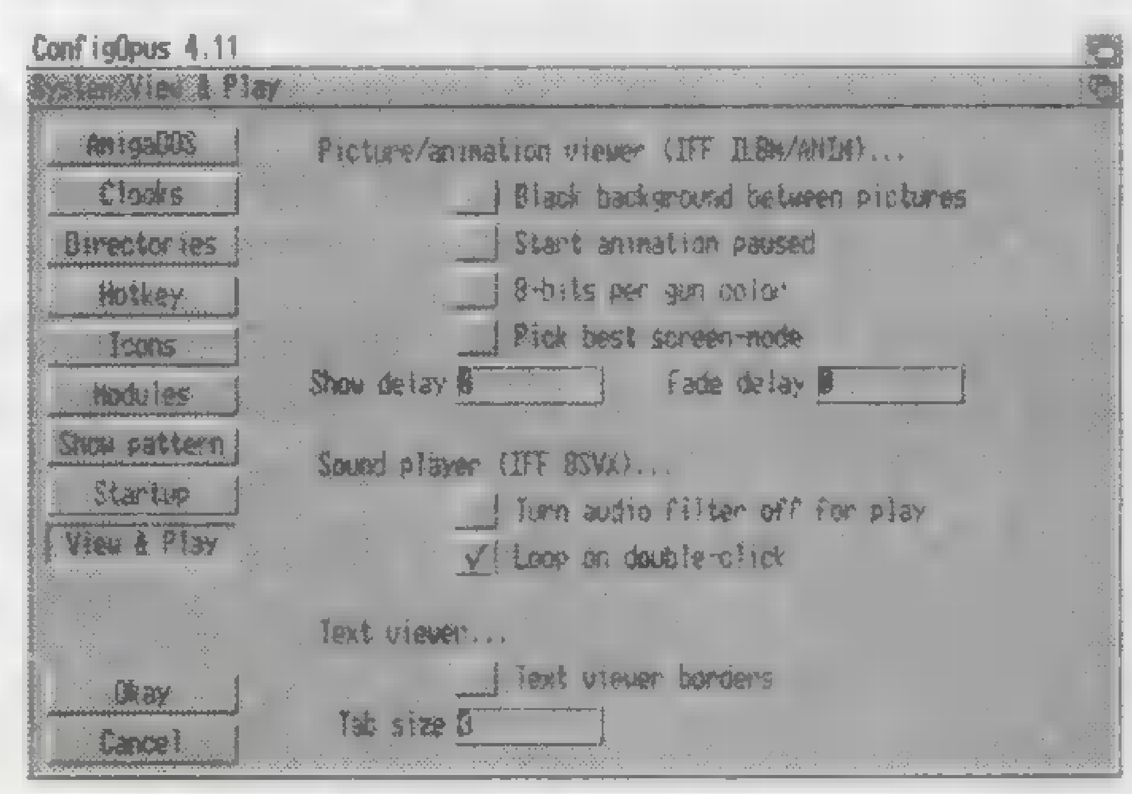


Zewnętrzne moduły programu.

Opcja Show pattern pozwala nam ustalić jakie pliki będą wyświetlane, a jakie nie. I tak można się pozbyć np. plików *.info, czy *.pic, ale jak już wspomniałem osobiście uważam, że warto widzieć wszystko, co jest na dysku. Niemiłe niespodzianki nikomu nie wsmak.

Opus pozwala na ustalenie tego, co się ma dziać z samym programem przy różnych okazjach. Do tego celu służą rozmaite skrypty ARexxa (programiki pisane w tym języku), które na nasze życzenie mogą być odpalane przy starcie programu, jego powrocie ze stanu zwinięcia (ikonifikacji), czy też z panelu konfiguracyjnego. Nazwy skryptów i ścieżki dostępu do nich ustalamy w opcji Startup.

Ostatnią opcją jaką chciałbym omówić jest ustawienie systemu wyświetlania rysunków, odgrywania dźwięków i przedstawiania tekstu. Amigowcy z żyłką artystyczną mają tu możliwość ustalenia np. opóźnień przy wyświetlaniu obrazów. Ładnie wygląda, ale potem staje się denerwujące. Najlepiej jest wybrać opcję: Pick best screen-mode. Jeśli chodzi o dźwięk to ustalamy czy filtr ma być wyłączany i czy chcemy by np. moduły, na które kliknęliśmy 2 razy zostały odgrywane w kółko (loop). Wyjście z pętli odbywa się przez wybranie polecenia Play "na pusto". Przy oglądaniu tekstu radzę ustalić wielkość tabulatorów na 3.



Muzyka z obrazem pomyka.

Myślę, że wystarczy Wam informacji by prawidłowo skonfigurować Opusa, zaś prawidłowe skonfigurowanie wystarczy, żeby owocnie z niego korzystać. Przy konfigurowaniu poznaliście bowiem już praktycznie wszystkie tajemnice programu i teraz wystarczy tylko przerzucać pliki z lewa na prawo, czy jak kto woli odwrotnie.

Wszelkie uwagi dotyczące artykułu proszę kierować na adres redakcji.

BRILLIANCE

Profesjonalny program graficzny cz.3

Tomasz Łoboda

To już trzecie spotkanie z niesamowitymi możliwościami Brilliance'a. Tym razem nadeszła pora, aby zająć się jednym z najważniejszych zastosowań komputerowej grafiki, czyli animacją obrazu. Opcje z tym związane są kolejną, niezwykle mocną stroną omawianego programu.

Jak wiadomo animacja składa się z szeregu następujących po sobie klatek (obrazków), z których każda zawiera minimalną zmianę położenia, kształtu lub koloru, reprezentowanych na niej obiektów. Takimi właśnie zasadami posługujemy się w programie "Brilliance". Chcąc stworzyć jakąkolwiek animację musimy przejść do "Animation menu", które oferuje nam następujące możliwości:

[1] "Load/Save animation" - umożliwia odczyt lub zapis gotowej animacji,

[2] "Set # of frames" - pozwala uaktywnić dodatkowe menu, a w nim:

- ***"Total frames"*** - określający ilość klatek animacji,

- ***"Add"*** - określa ilość klatek dodanych do klatki pierwszej, traktowanej jako klatka istniejąca standardowo,

- ***"Delete all"*** - kasuje wszystkie klatki animacji,

- ***"Delete frames {od} through {do}"*** - usuwa klatki animacji znajdujące się we wskazanym przedziale {od} - {do},

[3] "Add frame" - dodaje nową klatkę animacji na prawo od aktualnie edytowanej, kopiując jej zawartość,

[4] "Delete frame" - kasuje aktualną klatkę animacji,

[5] "Copy frames" - pozwala skopiować istniejące już klatki animacji na kilka różnych sposobów:

- ***"To all"*** - zawartość klatki aktualnej zostanie przeniesiona do wszystkich pozostałych klatek animacji,

- ***"To frames {od} through {do}"*** - aktualna klatka zostanie skopiowana do kilku innych następujących po sobie (określonych {od} - {do}),

- ***"Copy multiple frames {od} through {do} in front of frame {nr}"*** - wstawia wybrane klatki ({od} - {do}) przed klatkę o określonym numerze ({nr}),

[6] "Play range {od} - {do}" - odtwarza fragment animacji o ustalonych klatkach ({od} - {do}),

[7] "Frame selector" - pozwala wybrać użytkownikowi określoną klatkę animacji przy pomocy "suwaka",

[8] Pod wspomnianym wcześniej "suwakiem" znajduje się zespół przycisków o następujących funkcjach (patrząc od lewej strony):

- ***"First frame"*** - błyskawiczne, bezpośrednie przejście do pierwszej klatki animacji,

- ***"Previous frame"*** - umożliwia cofnięcie się o jedną klatkę wstecz,

- ***"Next frame"*** - powoduje przejście o jedną klatkę do przodu,

- ***"Last frame"*** - działa podobnie jak "First frame", przy czym przechodzimy do ostatniej klatki animacji,

- ***"Play loop backwards"*** - odtwarza animację w pętli począwszy od ostatniej klatki (w kierunku klatki pierwszej),

- ***"Play once backwards"*** - animacja zostanie odtworzona jedynie raz począwszy od ostatniej klatki,

- ***"Play ping-pong"*** - animacja zostanie pokazana od pierwszej do ostatniej klatki, a następnie od ostatniej do pierwszej itd.,

- ***"Play once forward"*** - jednokrotne powtórzenie animacji, start następuje od pierwszej klatki,

- ***"Play loop forward"*** - działa podobnie jak "Play loop backwards", rozpoczyna jednak od klatki pierwszej.

[9] "Animation speed" - pozwala regulować szybkość odtwarzania animacji (podając liczbę klatek wyświetlanych w ciągu sekundy),

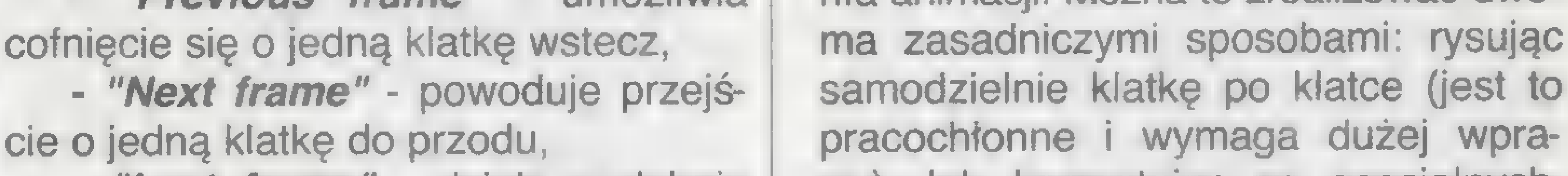
Warto w tym miejscu wspomnieć o tym, że w przypadku tworzenia animacji nowego znaczenia nabiera opcja "Clear". Wskazując tę ikonkę prawym przyciskiem myszy uzyskujemy menu "Clear frames", a w nim:

- ***"All frames"*** - zostaną "wyczyszczone" wszystkie klatki animacji,

- ***"Frames {od} through {do}"*** - usunięte zostaną obrazki ze wskazanych klatek animacji ({od} - {do}),

- ***"Current frame"*** - "wyczyszczona" zostaje tylko aktualnie wybrana klatka animacji.

Po ustaleniu ilości klatek pora przejść do bezpośredniego tworzenia animacji. Można to zrealizować dwoma zasadniczymi sposobami: rysując samodzielnie klatkę po klatce (jest to pracochłonne i wymaga dużej wprawy), lub korzystając ze specjalnych, dodatkowych opcji oferowanych przez program. Zajmijmy się tą drugą metodą. Jej podstawą jest umiejętne posługiwanie się brushami. Chcąc się tego nauczyć musimy najpierw zapoznać się z dwoma dodatkowymi panelami: **"Brush menu"** (przechodzi się do niego przyciskając prawy klawisz myszy na



Stencil menu.



Animations menu.

ikonce odpowiedzialnej za wycinanie brushy) i **"Stencil menu"** (uzyskujemy go wskazując prawym klawiszem myszy ikonkę "Stencil").

W **"Brush menu"** znajdziemy:

- ***"Load/save brush"*** - odczyt lub zapis aktualnego brusha,
- ***"Paste from clipboard"*** - wczytanie brusha z podręcznej pamięci (notatnika),
- ***"Copy clipboard"*** - zapis brusha do podręcznej pamięci,
- ***"Delete"*** - usuwa wskazany brush,

W centralnych "przegródkach" możemy równocześnie przechowywać maksymalnie do ośmiu brushy (w pamięci komputera oczywiście). Obok szybkiego dostępu do szeregu wyciętych fragmentów rysunku, możemy uzyskać niezwykle ciekawe efekty przy tworzeniu animacji.

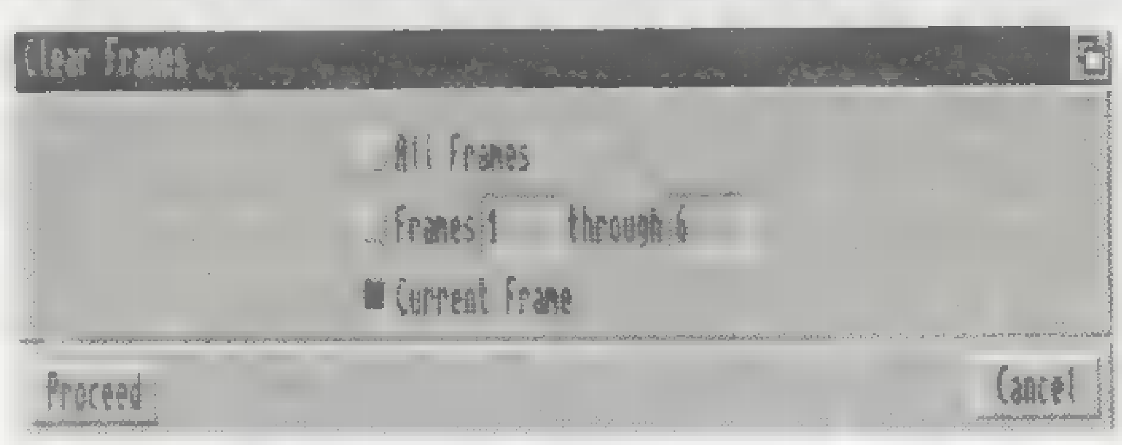
Dalej znajdują się:

- ***"Set handle upper left/right"*** - przesuwają miejsce uchwytu brusha pomiędzy lewym górnym, a prawym górnym rogiem,
- ***"Set handle lower left/right"*** - podobnie jak poprzednio, lecz wybieramy między lewym dolnym , a prawym dolnym rogiem,
- ***"Set handle center"*** - trzymamy wyciętego brusha w jego geometrycznym środku,
- ***"Adjust handle"*** - pozwala na dowolne uchwycenie wyciętego fragmentu (punkt, w którym uchwycimy brusha wskazujemy przy przyciśniętym lewym klawiszu myszy),
- ***"Use brush palette"*** - instaluje paletę brusha w miejsce aktualnej palety barw (uwaga: jeżeli wczytamy brusha, a na ekranie istnieje już jakiś rysunek, wybranie tej opcji może spowodować nieprzewidziane efekty, gdyż paleta brusha nie musi zgadzać się z aktualną paletą rysunku),
- ***"Remap brush to screen"*** - program dostosowuje kolory brusha do aktualnej palety rysunku,
- ***"Brush BGRND to FGRND"*** - przekazuje kolor spod prawego przycisku myszy w miejsce koloru przypisanego klawiszowi lewemu,
- ***"Swap BGRND with FGRND"*** - wymienia wskazany kolor w brushu na kolor tła i odwrotnie,
- ***"Set trans. color"*** - wymienia kolor tła w brushu na aktualny.

Ostatnie okienko zawiera kilka dodatkowych elementów pomocniczych dotyczących brushy (np.: ich wielkość w pikselach, ustawienie,itp.).

Drugim wspomnianym wcześniej panelem jest **"Stencil menu"** pozwalający na: dowolne operacje związane z wymianą pojedynczych kolorów na istniejącym już rysunku oraz zabezpieczenie wybranych barw przed jakąkolwiek zmianą. **"Stencil menu"** zawiera odpowiednio:

- ***"Invert stencil"*** - bezpośrednio wymienia ustawienie zabezpieczonych



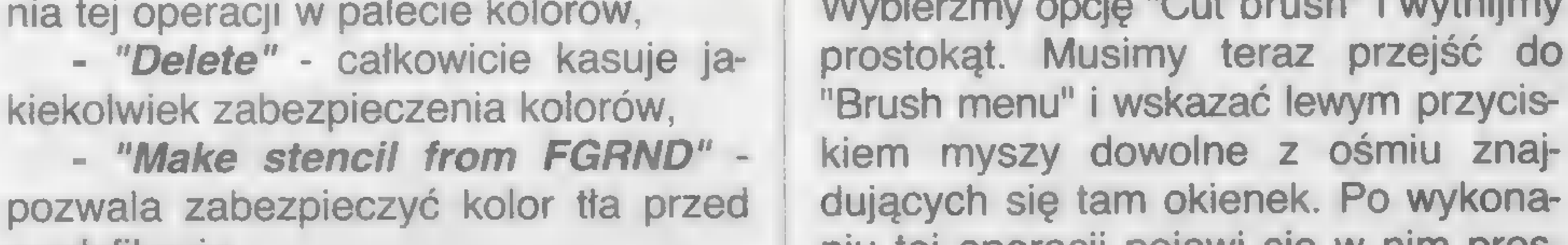
Clear frames.

kolorów na przeciwne, bez zaznaczenia tej operacji w palecie kolorów,
- ***"Delete"*** - całkowicie kasuje jakiekolwiek zabezpieczenia kolorów,
- ***"Make stencil from FGRND"*** - pozwala zabezpieczyć kolor tła przed modyfikacją,
- ***"Make stencil from colors"*** - zabezpiecza przed zmianą określone kolory,
- ***"Replace stencil"*** - wymienia zabezpieczone kolory między lewym, a prawym przyciskiem myszy,
- ***"Add to stencil"*** - dodaje nowy (odpowiednio zaznaczony kolor) do zabezpieczenia,
- ***"Subtract from stencil"*** - "zdejmuje" zabezpieczenie z zaznaczonego koloru,
- ***"Anim remake stencil"*** - po włączeniu tej opcji, wszystkie dokonywane przez nas zmiany będą dotyczyły całej animacji,
- ***"Clear set colors"*** - kasuje aktualne ustawienie zabezpieczonych kolorów,
- ***"Restore to entry"*** - powoduje powrót palety kolorów do stanu wyjściowego,
- ***"Undo last change"*** - przywraca stan palety barw sprzed ostatnio wykonanej czynności,
- ***"Select colors"*** - po wybraniu tej opcji wskazujemy kolor, który chcemy zabezpieczyć przed rysowaniem (kolor taki zostaje obwiedziony kółkiem w palecie barw),
- ***"Lasso color"*** - kolor, który chcemy zabezpieczyć zaznaczamy bezpośrednio na rysunku,
- ***"Invert set colors"*** - działa podobnie jak "Invert stencil" przy czym zmiana ustawienia kolorów zostaje uwidoczniona w palecie barw,
- ***"Show set colors"*** - pokazuje w palecie barw kolory, które są zabezpieczone przed zmianą.

Teraz już nic nie stoi na przeszkodzie, aby zająć się głównym tematem artykułu, czyli animacją. W przypadku stosowania brushy służą do tego celu dwa menu: "Anim brush menu" i "Tweening menu". Zajmijmy się najpierw panelem "Anim brush". Pozwala on na morphing brushy. Jego działanie najlepiej można prześledzić na prostym przykładzie.

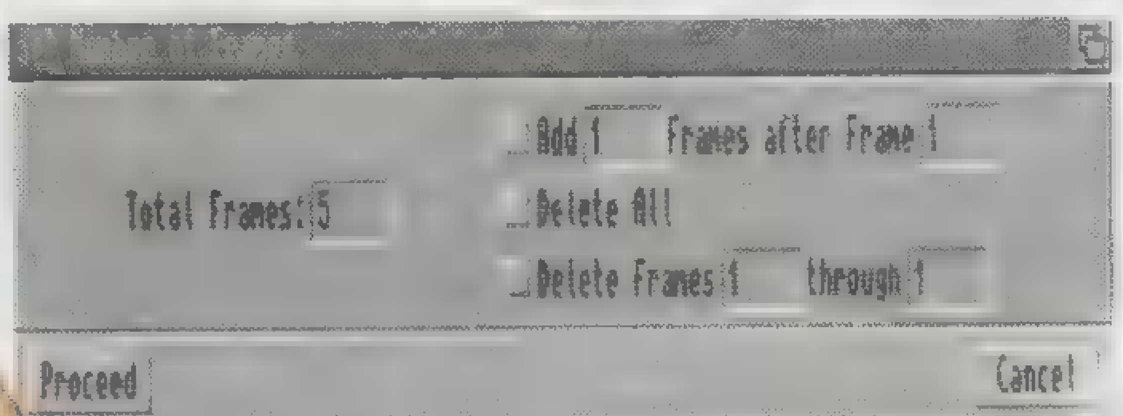
Zdefiniujmy dwie nieskomplikowane, wypełnione figury: koło i prostokąt. Wybierzmy opcję "Cut brush" i wytnijmy prostokąt. Musimy teraz przejść do "Brush menu" i wskazać lewym przyciskiem myszy dowolne z ośmiu znajdujących się tam okienek. Po wykonaniu tej operacji pojawi się w nim prostokącik symbolizujący wycięty wcześniej brush. Podobnie postępujemy z kołem. Następnie przechodzimy do "Anim brush menu" i tutaj należy się kilka dodatkowych wyjaśnień:

- w pierwszej części tego panelu znajdują się podstawowe operacje typu: ***"Load/save anim brush"*** (czyli odczyt i zapis stworzonej animacji brusha) oraz ***"Delete"*** (usuwa animację),
- dalej znajduje się okienko informujące nas o ilości dostępnych klatek animacji w "Animation menu" (jak się dalej przekonamy jest to dość ważne) oraz o sposobie jej realizacji (***"Cut backward/forward"***),



Brush menu.

- okno ***"Morph"*** pozwala na określenie: ilości klatek, w których ma nastąpić zmiana kształtu brusha (ilość tak zdefiniowanych przejść wcale nie musi zgadzać się z ilością klatek animacji w "Animation menu") oraz między

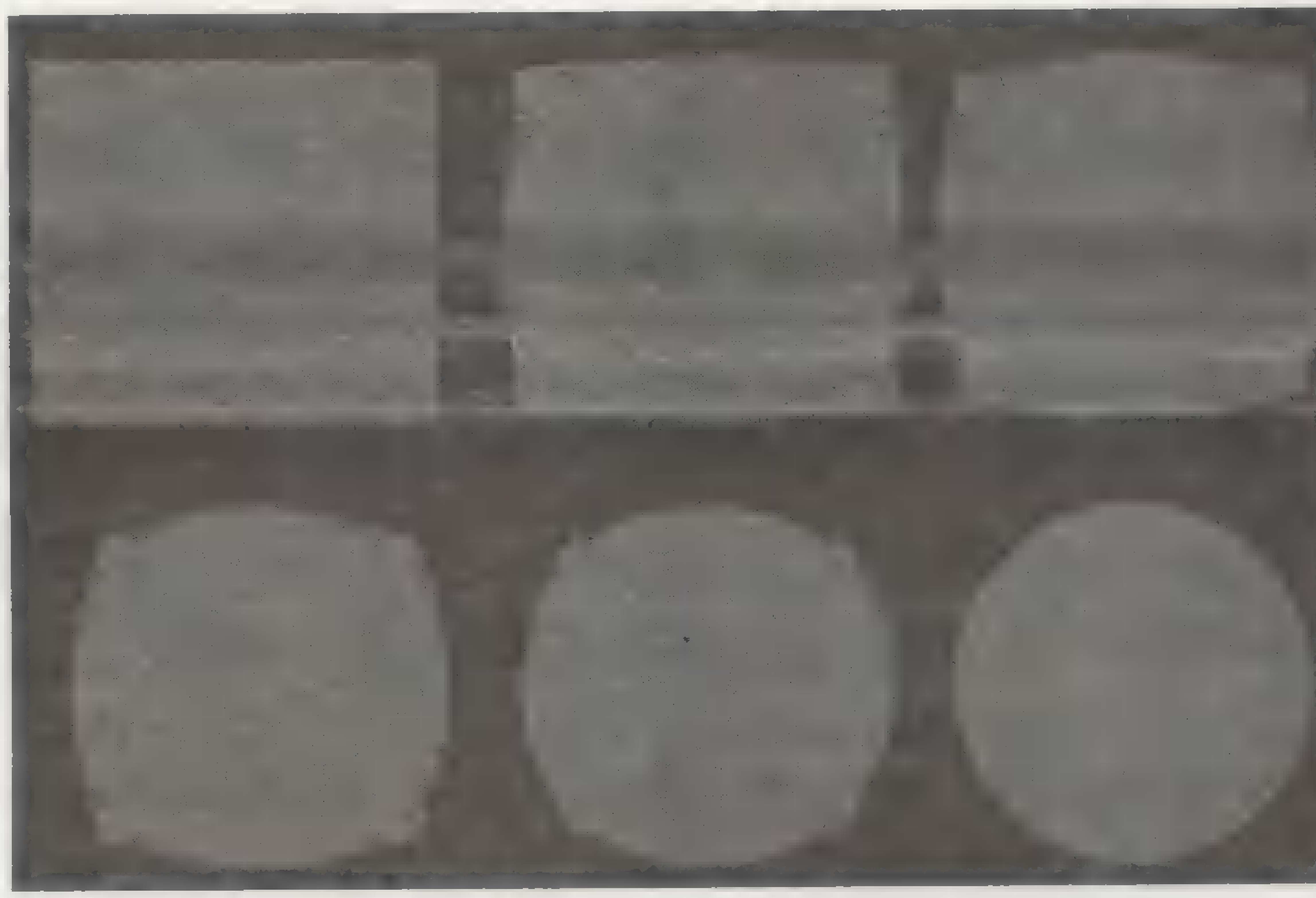


Set frames.

jakimi brushami to nastąpi (jak łatwo zauważyć, ułożenie przycisków w opcji "Morph from brush well" odpowiada ustawieniu okienek na brushu w "Brush menu"),

- rola poszczególnych przycisków w ostatniej części panelu "Anim brush menu" jest analogiczna do działania podobnych klawiszy w "Animation menu" (przyciski te pozwalają na przejścia pomiędzy poszczególnymi etapami morphingu brusha oraz określają sposób następowania po sobie kolejnych klatek tej animacji).

Ustawmy ilość klatek animacji brusha na pięć i wybierzmy uaktywniony numer w "Morph from brush well". Teraz wystarczy uzbroić się w odrobinę cierpliwości czekając na wykonanie przez program całości morphingu. Kiedy obliczenia dobrną do szczęśliwego końca, możemy obejrzeć poszczególne klatki przy pomocy wspomnianych wcześniej przycisków znajdujących się w ostatnim oknie "Anim brush menu". Chcąc uzyskać pełną animację należy przenieść ją do "Animations menu" wykorzystując do tego celu opcję "Grid lock".



Kolejne klatki morphingu brusha.

Postępowanie jest wówczas następujące:

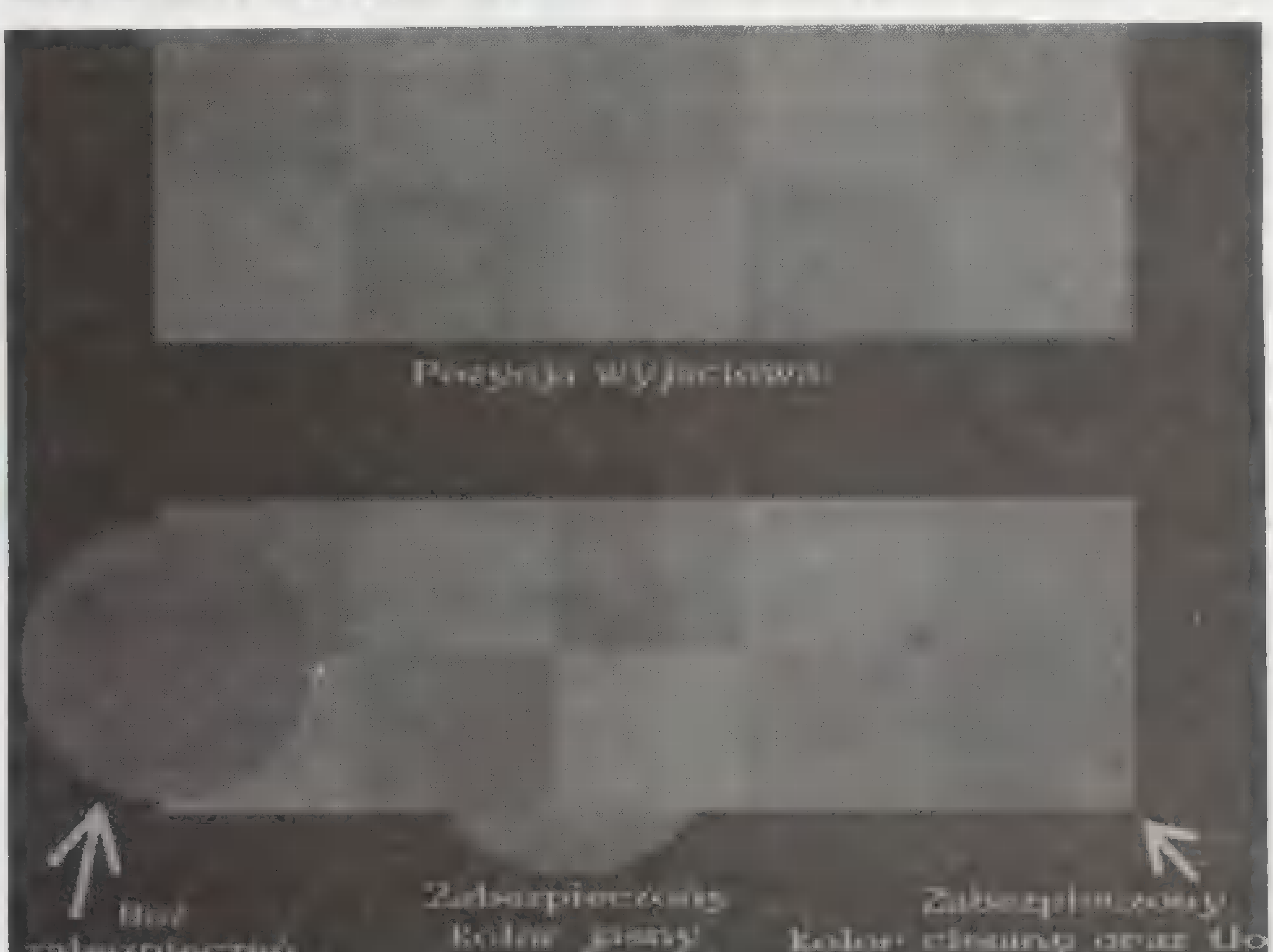
- przechodzimy do "Animations menu" i ustalamy ilość klatek na pięć,

- wybieramy opcję "Grid lock" i powiększamy za jej pomocą wartość skoków kursora (np. 30 na 30 punktów - może to nam w precyzyjnym ustawieniu położenia brusha w poszczególnych klatkach animacji),

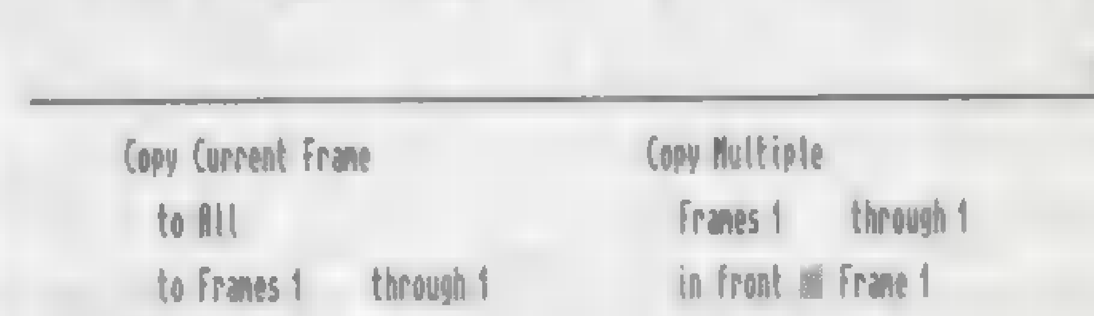
- ustawiamy pierwszą klatkę w "Animations menu" oraz pierwszy kształt w "Anim brush menu" ("First frame"),

- korzystając z koordynatów położenia kursora ustawiamy pierwszego brusha w wybranym przez nas miejscu. Po przyciśnięciu lewego przycisku myszy zostanie narysowany pierwszy kształt brusha, a program automatycznie wywoła drugą klatkę morphingu, którą w analogiczny sposób umieszczamy w drugiej klatce animacji. Podobnie postępujemy z pozostałymi klatkami.

W opisany powyżej sposób wykonaliśmy najprostszą animację, którą najlepiej obejrzeć korzystając z opcji "Play ping-pong". Nie jest to może zbyt



Przykład działania opcji "Stencil".



Copy frames.

wielkie osiągięcie, lecz miało ono na celu jedynie wprowadzenie Czytelników w zagadnienia związane z animacją komputerową.

To tyle, jeżeli chodzi o trzecią część opisu "Brilliance'a". W kolejnym odcinku przedstawię ostatnie opcje i menu oferowane przez program oraz postaram się przybliżyć ich praktyczne wykorzystanie w połączeniu z wcześniej opisywanymi możliwościami. ❑

PRZYSZŁOŚĆ

PRZESZŁOŚĆ

Minęły już dwa miesiące 1994 roku, można więc ze spokojem podsumować ubiegłoroczne działania różnych firm zaopatrujących użytkowników Amigi w sprzęt i oprogramowanie.

Na początek firma Commodore. Z pewnością 1993 rok nie był dla niej najlepszy. Główną przyczyną złego samopoczucia ludzi, którzy kierowali Commodore były kłopoty finansowe firmy. Problemy te doprowadziły do zamknięcia oddziałów przedsiębiorstwa w kilku krajach (w Polsce również). O problemach finansowych firmy Commodore zadecydowały głównie dwa czynniki. Pierwszym z nich był fakt zainwestowania olbrzymich pieniędzy w produkcje klonów IBM. Commodore nie było przygotowane do walki z dalekowschodnią konkurencją w związku z tym spodziewane zyski z rynku PC nigdy nie nadeszły Drugim powodem ciągłego powiększania się długów firmy było pozostawienie przez długi czas rynku Amigi samemu sobie. Przez okres kilku lat (praktycznie od momentu pojawienia się A3000) komputery te poddawano jedynie kosmetycznym zmianom (tzn. wymieniano obudowy - A600 lub system operacyjny - A500+) cały ciężar rozwoju techniki pozostawiono na barkach producentów dodatkowych peryferii (z czego z resztą ci ostatni wywiązali się całkiem nieźle). Efektem takiego działania był przestarzały sprzęt. W momencie, gdy wszyscy ważni producenci zdążyli już zapomnieć o technologiach 16-bitowych, Commodore nie miało jeszcze pomysłu na tanią 32 - bitową Amigę. Na szczęście (dla ami gowców) nadszedł kryzys finansowy i nasz kochany producent zrozumiał, że nie ma czego szukać na rynku PC oraz, że prawdziwe pieniądze można zrobić również sprzedając Amigi. Finanse sprawiły, że potrzebny był szybki sukces. I cokolwiek złego moglibyśmy powiedzieć o ludziach kierujących firmą trzeba im przyznać, że sukces ten udało im się osiągnąć. Postawienie na Amigę opłaciło się. W ciągu zaledwie kilku miesięcy udało się rozpocząć produkcję komputerów, których nie można było wykonać przez kilka lat, a które bardzo szybko stały się hitami. Pojawienie się Amigi 1200, 4000 oraz CD 32 spowodowały coś w rodzaju przełomu. Przyczyny wydają się stosunkowo proste. Komputery te w końcu dorównały konkurencyjnemu sprzętowi spod herbu PC i Macintosha. Niebagatelny też był fakt sporego obniżenia cen. Commodore zrozumiało, że oprócz komputerów bardzo prostych jak Amiga 500, 600 i bardzo szybkich jak np. A3000 istnieje także spora grupa ludzi, którym proste modele nie wystarczają, a na drogie ich nie stać. Dokładna analiza potrzeb, którą Commodore powinna wykonać już dawno stała się podstawą do następujących decyzji:

1. rezygnacja z technologii 16-bitowej - co oznacza koniec produkcji Amigi 600;

2. modelem podstawowym przeznaczonym do zastosowań domowych staje się Amiga 1200;

3. nałogowi gracze, których nic innego nie interesuje mogą zaopatrzyć się w CD32;

4. najbogatsi użytkownicy mogą zaś kupić sobie Amigę 4000/040;

5. natomiast ludzi środka na pewno zainteresuje Amiga 4000/030.

Aby nie popadać w zbytni optymizm trzeba stwierdzić, że Commodore popełnił jednak kilka błędów. Są to pomyłki natury technicznej. W przypadku modelu 4000 była to całkowita rezygnacja z kontrolera SCSI, natomiast w przypadku Amigi 1200 rezygnacja z zastosowania stacji dysków wysokiej gęstości. Przykrą niespodzianką dla wielu użytkowników była także wiadomość o odsunięciu realizacji projektu DSP (16 - bitowego dźwięku). Dało się także zauważyć pewną irytację, którą wywołały układy AGA. Są one na pewno krokiem naprzód, ale właściciele komputera graficznego oczekiwali czegoś więcej. Głównym powodem nerwów jest fakt wyposażenia 32 - bitowych układów AGA w 16 - bitowy blitter (odpowiedzialny za przenoszenie danych graficznych). Takie rozwiązanie oznacza spowolnienie operacji graficznych.

Nadeszła już pora, aby zbadać zamiary firmy Commodore w 1994 roku. Najważniejszym problemem, którym będzie się ona zajmowała na początku roku będzie zakończenie projektu Amigi 4000 T (Tower). Jest to stosunkowo niewielkie zadanie dla tak dużej firmy. Reszta zamiarów firmy jest na razie oficjalnie nieznane. Opieramy się jedynie na pewnych przeciekach, o których nie można mówić jak o wiarygodnych informacjach. Wszystkie te informacje są spójne w kilku punktach. Pierwszym z nich jest fakt projektowania nowych układów graficznych określanych jako AAA. Mają być one pięciokrotnie szybsze od obecnie stosowanych AGA. Powinny umożliwiać wyświetlanie 16.8 milionów kolorów dysponować rozdzielczościami rzędu 1200 na 1024



punkty. Układy AAA mają być także wyposażone w możliwość wykonywania obliczeń dla potrzeb programów 3D (Imagine, Real 3D). W momencie powstawania tego artykułu nadeszła informacja o rozpoczęciu testów tych układów.

Nieco mniej informacji dotyczy komputera, w którym Commodore ma zainstalować te układy. Mówi się o nowych Amigach wieloprocesorowych lub też opartych o procesory RISC (stosowane w dużych stacjach graficznych). Najbardziej wiarygodne są informacje o Amidze, której sercem będą dwa procesory 68040 (lub następcy tych układów) oraz drugim modelu opartym o procesor RISC oraz graficzne układy AAA. Mówi się też o zastąpieniu starego dobrego Workbencha czymś w rodzaju wielosystemowego odpowiednika. W efekcie mamy otrzymać maszynę, na której oprócz aplikacji napisanych specjalnie dla Amigi będzie można używać programów dla środowiska Windows NT (IBM PC) czy też systemu Macintosha. Na dodatek zastosowanie procesora RISC ma spowodować, że aplikacje te będą pracowały kilkakrotnie szybciej niż np. w maszynach opartych o nowy procesor firmy Intel (Pentium - 80586). Trudno jednak powiedzieć czego doczekamy się w rzeczywistości.

Czas "dobrać" się do innych firm. Niestety tutaj nie dysponujemy praktycznie prawie żadnymi danymi. Wiadomo jedynie, że najprawdopodobniej niektóre spośród nich rozszerzą swoją działalność o inne komputery niż Amiga (kłopoty finansowe Commodore są tu głównym powodem). Na pewno wierne temu rynkowi pozostanie ASDG (producent oprogramowania i hardwaru graficznego) oraz GVP, na którą to firmę Commodore zrzucił produkcję urządzeń peryferyjnych do Amigi. Niestety nie można było uzyskać informacji o nowych produktach, które będziemy mogli nabyć w 1994 roku (no cóż - konkurencja czuwa). Najczęstszą odpowiedzią była informacja, że firma ma zamiar kontynuować rozwój któregoś ze swoich produktów. W większości wypadków oznacza to rozpoczęcie produkcji urządzeń, których reklamy ukazywały się na łamach zachodnich czasopism już w ubiegłym roku. Podobnie sprawa wygląda w przypadku oprogramowania. Firmy współpracujące z Commodore w produkcji przeznaczonej dla Amigi wykazują dużą ostrożność. W ten właśnie sposób informują o swoich zamierzeniach takie firmy jak ASDG (Art Department, Morph Plus, True Print 24), Activa (Real 3D), Europress Software (AMOS), Electronic Arts (Deluxe Paint), Impulse (Imagine), INOVAtronics (Directory Opus, Can Do), Oxxi - Ageis (Sbase, Video Stage). Nie oznacza to całkowitej rezygnacji z nowych projektów. Colanto zamierza w znaczący sposób rozbudować Personal Paint'a, który ma stać się programem o możliwościach bliższych profesjonalnym programom graficznym w rodzaju Image FX czy też Art Department. Najszerszą ofertę nowych produktów ujawniła Digita International na początek tego roku zapowiedziano (i po części zrealizowano) następujące projekty Wordworth 3, GTI, Money Matters 3.1, Digita Print Manager 2. Firma Gold Disk zamierza zrealizować nowy projekt. Po sukcesie nowej wersji pakietów DTP (Professional Page 4 i Pagesetter 3) programiści zamierzają zabrać się za inny produkt, a mianowicie Video Directora i wykonać nową wersję tego programu do tworzenia obrazów wideo. Natomiast firma Helios znana dotychczas w Wielkiej Brytanii w obszarze zastosowań edukacyjnych postanowiła opracować konkurencyjny w stosunku do AMOSa język programowania o nazwie HeliOS zorientowany głównie na produkcję gier. Podobnie z nowym językiem programowania w rok 1994 chce wejść Hisoft oddając do rąk użytkowników nowe wersje Hisoft Basica oraz zapowiadając produkcję 16-bitowych programów do obsługi samplerów oraz aplikacji do obróbki obrazu. Na zakończenie jeszcze dwie firmy. Virtual Reality Labs w nowym roku chce zaproponować nowe wersje Visty Pro oraz Distant Suns, a z nowych projektów proponuje program Vista Morph, który umożliwia wykonanie łagodnego przejścia pomiędzy dwoma krajobrazami wykonanymi przy pomocy Visty Pro. No i dobrze znany w Polsce Softwood. "Niestety" oni niczego nie zapowiadają - oni produkują, ich produkt na rok 1994 nosi nazwę Final Writer.

I w ten sposób dobrnęliśmy do końca. Na ile nasze informacje okazały się prawdziwe okaże się w grudniu. Mamy jednak nadzieję, że popełniliśmy błąd w ocenie rynku i będzie on znacznie bardziej nasycony nowym sprzętem i oprogramowaniem niż to wynika z zapowiedzi firm. ❑



Z ŻYCIA KOMPUTEROWEGO PSA, CZYLI LET'S SWING IT.

[illegible] ynski

Niniejszym artykułem chciałbym zapoczątkować cykl polemik na różne komputerowe tematy. Każdy może nadsyłać tu swoje uwagi. I Ci, którzy się zgadzają i Ci, których zajmują odmienne stanowiska. Dyskutować może każdy.

Będziemy przestrzegać jednej zasady: żadnego rzucania mięsem i innych takich pseudoargumentów. Bo tam gdzie argumenty zastąpione zostają kulami, źle kończy się to dla wszystkich. Zaczniemy oczywiście z grubej rury...

Ostatnio pewien mój kolega poprosił mnie o przesłanie mu animacji z pieskiem, która była zamieszczona na jednym z dysków PD AMIGOWCA (2/93). Prośbę spełniłem chętnie i muszę przyznać, że dawno nic tak miłego nie wpadło mi do ręki. W trakcie oglądania owej animacji przyszło mi do głowy, że można by w jej tle puścić sobie jakąś muzyczkę. Z komputera oczywiście. Wybrałem melodyjkę "Let's swing it". Potem odpaliłem sobie Opusa, by coś skopiować, PowerPackera, aby coś trzeba było przepakować i CEDa, gdyż chciałem napisać kilka słów listu do kolegi. Na koniec wpadłem na pomysł, że można by tę animację uruchomić jeszcze raz. Udało się. Po tych wszystkich operacjach okazało się, że procesor mojej Amigi jest zajęty tylko w 60%...

Próbowaliście kiedyś robić kilka rzeczy jednocześnie na komputerze klasy IBM? Jeśli nie to nie macie czego żałować. Hasło "Intel Outside" w pełni oddaje rzeczywistość takiego procesu. Coś lub ktoś tu po prostu nie nadąża i "wysiada" (outside).

Wielu z nas, zapalonych amigowców, niejednokrotnie toczyło zażarte spory o to jaki komputer jest lepszy: Amiga, czy IBM, a może Macintosh? Przeważnie z tych sporów nic nie wynikało. Czasami najwyżej jakaś koszula uległa naderwaniu w trakcie żywego wymachiwania rękoma. A nie chcielibyście wiedzieć jak to wygląda faktycznie? Tak zupełnie obiektywnie? Dlaczego takie porównania nigdy nie wychodzą?

Zacznijmy od klasy sprzętu. Tu pojawia się pierwszy kłopot. Amigi o niższych numerach nie są standardowo wyposażone w twarde dyski. Oczywiście znacznie obniża to cenę, ale też uniemożliwia uruchomienie niektórych, bardziej profesjonalnych programów. Z drugiej strony dla całej rzeszy komputerowej młodzieży Amiga 500, czy 600 staje się wspaniałym narzędziem, dzięki któremu można poznać fascynujący świat komputerów. Czasami "zabawa" kończy się na grach, ■ przez to Amiga zyskuje miano komputera tylko do gier, co oczywiście prawdą nie jest. Mści się to na tych, którzy chcą zacząć używać swojej maszynki do poważniejszych zastosowań. Zakup twardego dysku i dodatkowej pamięci drastycznie podbija cenę takiej np. A500, ■ jej szybkość nagle staje się wysoce niewystarczająca. Nie tędy droga Drodzy Państwo! Owszem - na tak wyposażonym komputerze można wcale wydajnie pracować, tworzyć grafiki, teksty, czy programy. Jeśli dodatkowo w pewnym momencie będzie nas stać na kartę turbo, to zniknie wiele zastrzeżeń. Jednak mimo wszystko ciągle nie będzie to zestaw porównywalny z rozbudowanymi komputerami typu IBM 486, czy Macintosh Quadra. Tu porównanie trzeba zacząć od drugiej strony. Od Amigi 4000/040. Jest to ■ tej chwili najlepszy dostępny model tego komputera. Dodatkowo można go wyposażyć w karty graficzne, dźwiękowe i inne ciekawostki, tak że mucha nie siada...



W tym miejscu porównanie należy przenieść na płaszczyznę oprogramowania. Obecnie nadszedł taki czas, że Amiga przestaje ustępować pola w czymkolwiek. Nowe wersje programów, które właśnie pojawiają się na rynku są nie tylko porównywalne z ich odpowiednikami na IBMa, czy Macintosha, ale wręcz niedościgłe. Nowa wersja programu PageStream według naszych opinii nie ma odpowiednika wśród największych gigantów DTP. Jej cena zaś jest tylko ułamkiem ceny tamtych programów, co jest szczególnie istotne w dobie oryginałów. Podobnie rzecz się ma w grafice bitowej. DeLuxe Paint, Brilliance, ImageFX oraz wiele innych to programy nie dające sobie łatwo utrzeć nosa. Tworzenie "rzeczywistych" światów wprost z naszej wyobraźni osiągnęło na Amidze szczyty perfekcji. W dziedzinie video komputer ten przestaje ustępować nawet profesjonalnym stacjom graficznym. Najlepszym przykładem są sceny do filmów Jurassic Park, czy Terminator. Zrobienie na Amidze wielu "ciekawszych" scen do Gwiezdnych Wojen też obecnie nie byłoby żadnym problemem. Oczywiście nie każdy musi tworzyć grafiki komputerowe. Amiga ma się dobrze również w dziedzinie pisania tekstów, ■ przykładem niech będzie FinalWriter, czy "stary", poczciwy CED (program ten wbrew pozorom nie ma swojego odpowiednika na innych komputerach). DynaCAD, program przydatny projektantom, wg opinii swoich twórców jest najlepszym programem tego typu w ogóle. Jeśli nawet jest on równie dobry jak najlepsze programy, to i tak to wystarczy. Pewną luką Amigi są arkusze kalkulacyjne. Wprawdzie istnieje ProCalc. Program szybki i o wielu możliwościach, jednak nie jest on tak rozbudowany jak np. EXCEL. W krótkim czasie i tu powinny nastąpić zmiany.

Poza "gigantami" należy wspomnieć o całej masie programów edukacyjnych czy pomocniczych, których na Amigę powstała taka masa, że tylko wybierać. Znakomita większość programów na Amigę jest po prostu tania! "Szczególnie" tanie są programy PD, czy Shareware. A jeśli ktoś miałby zastrzeżenia do ich poziomu, to wypada przytoczyć przykład programu służącego do obsługi modemu: Term. Nie ma programu o takich możliwościach na IBMa!

AMIGOWCY!

Wasz komputer jest naprawdę dobry! ❑

ANKIETA AMIGOWCA

Drogi Czytelniku !

Po raz kolejny dajemy Tobie szansę wyżycia się na nas. Narzędziem, które chcemy abyś w tym celu wykorzystał jest nasza druga ankieta (zapominalskim przypominam, że pierwsza była zamieszczona w czerwcowym numerze Amigowca z 1992 roku).

Pomysł powtórnego zamieszczenia w Amigowcu narzędzia czytelniczej krytyki wziął się z przekonania, że przeszliśmy już pewien okres rozwoju naszego pisma. Czas ten posłużył nam do ustalenia, jakiego rodzaju pismo chcemy wydawać, kto czym powinien się zajmować. Oprócz tego budowaliśmy całe techniczne zaplecze, bez którego Amigowiec byłby jedynie gazetką o zasięgu co najwyżej szkolnym. Teraz dysponujemy już instrumentami i ludźmi, których możliwości zapewnią naszym produkcjom utrzymanie w miarę równego poziomu oraz rokują na równomierny rozwój. Mamy nadzieję już nigdy więcej nie popełniać grzechu, który często był nam wytykany, a polegającego na tym, że Amigowiec ukazywał się na rynku niecyklicznie.

Pozbycie się problemów "technicznych" pozwoliło nam zająć się lepiej zawartością czasopisma. Zaowocowało to pojawieniem się nowych działów oraz wpłynęło na zwiększenie ilości stron do czytania. Niemniej jednak doszliśmy do wniosku, że dokonywanie wyboru tematów do Amigowca nie może zostać oparte jedynie o doświadczenia członków redakcji oraz serwisy informacyjne zachodnich firm. Taka postawa spowodowałaby, że stracilibyśmy kontakt z polską rzeczywistością. Problem postawiony w ten sposób narzucił nam metodę postępowania. W naszej nowej ankiecie można spotkać trzy rodzaje pytań. Pierwszy spośród nich można by nazwać "historycznym", chcemy dzięki nim przekonać się na ile Amigowiec w ciągu ostatniego roku spełniał Wasze oczekiwania. Druga grupa pytań pomoże nam ustalić do czego służy Wam Amiga, a więc jakich informacji od nas oczekujecie. Ostatnia podgrupa pytań posłuży nam do ustalenia aktualnego "poziomu technicznego" w Waszych domach. Odpowiedzi, których udzielicie z pewnością będą miały wpływ na wybór sprzętu do testów.

Teraz parę słów o technice wypełniania ankiety. Na pytania, przy których widnieją kwadratowe pola odpowiadamy stawiając krzyżyk. UWAGA!!! W odpowiedzi na pierwsze pytanie należy postawić TYLKO jeden taki znaczek. Jeżeli ilość miejsca, którą pozostawiliśmy Tobie do dyspozycji przy pytaniach o charakterze opisowym okaże się niewystarczająca, możesz odpowiedzieć na osobnej kartce papieru, którą dołączysz do ankiety (najlepiej spiąć). Jeżeli uważasz, że ankieta nie obejmuje wszystkich problemów, którymi Twoim zdaniem powinniśmy się zająć, możesz przysłać nam uzupełnienie w formie listu, który z pewnością uhonorujemy. Jednocześnie nasza akcja jest właściwym momentem dla tych wszystkich, którzy chcieliby rozpocząć współpracę z naszym czasopismem. Jeśli jesteś zainteresowany taką współpracą możesz nas o tym poinformować w swoim liście, mile widziane będą także materiały do natychmiastowego wykorzystania.

Na zakończenie jeszcze kilka zdań o motywacji. Tak jak w przypadku poprzedniej ankiety postanowiliśmy wśród czytelników, którzy wyślą nam wypełniony formularz rozlosować kilka cennych nagród. Nagrodami będą między innymi:

* gry komputerowe
* joysticki
* dyskietki

Oczywiście najwyżej honorowane będą te spośród odpowiedzi, które oprócz samego formularza będą zawierały uwagi lub propozycje Czytelnika rozszerzające problematykę ankiety.

Życzymy wszystkim przyjemnego wypełniania. Tylko nie zapomnijcie wpisać swojego adresu lub telefonu.

1. Czy Amigowiec spełnia twoje oczekiwania?

☐ tak
☐ raczej tak
☐ raczej nie
☐ nie

2. Jakiego rodzaju artykuły czytasz najczęściej?

☐ depesze i nowinki
☐ recenzje
☐ dokładne opisy
☐ artykuły problemowe (warsztaty)
☐ publicystykę

3. Jakiego rodzaju zastosowania Amigi powinny stanowić główny temat artykułów w Amigowcu?

☐ grafika
☐ wideo
☐ edycja tekstów i DTP
☐ bazy danych i arkusze kalkulacyjne
☐ programowanie
☐ inne (wymień jakie) ....

4. Wymień najlepsze twoim zdaniem artykuły zamieszczone w Amigowcu do numeru 1/94 włącznie?

....
....
....
....
....

5. Wymień autorów, których styl pisania najbardziej Ci odpowiada?

....
....
....
....
....
....

6. Co Twoim zdaniem należałoby zmienić w treści Amigowca?

....
....
....
....
....
....
....
....
....
....
....

7. Jakie zmiany należałoby wprowadzić w szacie graficznej Amigowca?

....
....
....
....
....
....
....
....
....

8. Która okładka Amigowca najbardziej Ci się podobała?

....

9. Czy uważasz, że Amigowiec jest łatwo dostępny w pobliżu Twojego miejsca zamieszkania?

☐ tak
☐ nie

10. Gdzie kupujesz Amigowca?

☐ w kiosku Ruchu
☐ w sklepie komputerowym
☐ w księgarni
☐ inne ....

11. Od kiedy czytasz Amigowca (miesiąc i rok)?

....

12. Jakie inne pisma komputerowe czytasz?

....
....

13. Jaki posiadasz komputer?

☐ Amiga 500/600
☐ Amiga 1200
☐ Amiga 2000
☐ Amiga 3000
☐ Amiga 4000/030
☐ Amiga 4000/040
☐ CDTV/CD32

14. Jakie posiadasz urządzenia peryferyjne?

☐ monitor monochromatyczny
☐ monitor kolorowy
☐ stacje dysków
  ☐ 3.5"
  ☐ 5.25"
☐ dysk twardy o pojemności .... MB
☐ dodatkową pamięć RAM o pojemności .... MB
☐ drukarka
☐ skaner
☐ digitizer
☐ genlock
☐ inne (wymień jakie)..

15. Jak długo posiadasz komputer?

....

16. Do jakich prac używasz swoją Amigę?

☐ grafika
☐ wideo
☐ edycja tekstów i DTP
☐ bazy danych i arkusze kalkulacyjne
☐ programowanie
☐ inne (wymień jakie) ....

17. Gdzie używasz Amigę?

☐ w domu
☐ w pracy

18. Wymień nazwy najczęściej wykorzystywanych programów?

....
....
....
....
....

19. Uwagi dodatkowe

....
....
....
....
....
....
....
....
....
....
....

Dane osobiste

imię ....

nazwisko ....

wiek ....

adres: ....
....
....

KURS ASEMBLERA

Adam Gragrowicz

Zgodnie z zapowiedzią z poprzedniego odcinka kursu dzisiaj dowiecie się co robi rozkaz BTST i inne instrukcje należące do tej samej grupy. W następnym numerze chcę Wam zaproponować wspólne napisanie programu bardzo prostego kalkulatora. Już teraz, aby przygotoywać się do tego zadania, po opisie rozkazu BTST (i jemu podobnych) przedstawię instrukcje wykonujące operacje matematyczne.

Rozkaz BTST należy do grupy rozkazów wykonujących operacje na pojedynczych bitach. Za pomocą instrukcji z tej grupy mamy możliwość testowania lub zmiany stanu poszczególnych bitów komórki pamięci lub rejestru.

BTST - Test a bit. (Test bitu.)

Zapis: BTST Dn,przeznaczenie lub BTST #n,przeznaczenie

Rozkaz BTST testuje określony bit w operandzie przeznaczenia. Jeśli testowany bit jest wyzerowany, to BTST ustawia na 1 znacznik Z. W przeciwnym wypadku (gdy testowany bit jest ustawiony) znacznik Z jest zerowany. Numer bitu, który ma być testowany, może być zawarty w rejestrze danych (zapis BTST Dn,...) lub być określony w sposób bezpośredni (zapis BTST #n,... ; n - numer bitu do przetestowania). Bity numerowane są od zera, które oznacza najmniej znaczący bit. W przypadku testowania bitów rejestru danych rozkaz BTST ma rozmiar długiego słowa, co oznacza, że numer bitu do testowania zawiera się w granicach od 0 do 31. Natomiast, gdy testowane są bity w komórce pamięci, BTST ma rozmiar bajtu, czyli sens mają jedynie numery bitów od 0 do 7.

Znaczniki:

X - Nie zmieniany.

N - Ustawiany, gdy testowany bit jest wyzerowany. W przeciwnym razie zerowany.

Z - Nie zmieniany.

V - Nie zmieniany.

C - Nie zmieniany.

Przy stosowaniu rozkazu BTST dozwolone są następujące tryby adresowania:

```
BTST ...,Dn)
BTST ...,(An)
BTST ...,(An)+
BTST ...,-(An)
BTST ...,p(An)
BTST ...,p(An,Rm.W)
BTST ...,p(An,Rm.L)
BTST ...,adr.W
BTST ...,adr.L
BTST ...,p(PC)
BTST ...,p(PC,Rn.W)
BTST ...,p(PC,Rn.L)
```

W przypadku rozkazu BTST z numerem bitu do testowania zawartym w rejestrze danych, możliwe jest testowanie bitów w liczbie określonej w sposób bezpośredni. Przykładowo BTST D0,#128 przetestuje w liczbie 128 bit o numerze zawartym w rejestrze D0. W praktyce takie zastosowanie rozkazu BTST jest pozbawione najmniejszego sensu. Pragnę podkreślić, że liczba określona bezpośrednio jest traktowana tak jak komórki pamięci, czyli może mieć rozmiar bajtu.

Zauważmy, że nie ma możliwości testowania bitów rejestrów adresowych. Aby sprawdzić stan jakiegoś bitu rejestru adresowego należy najpierw ten rejestr przepisać do rejestru danych (np. instrukcją MOVE.L An,Dm).

Przykład: BTST #5,10(A1,D6.L) powoduje przetestowanie stanu piątego bitu w komórce pamięci o adresie obliczonym w następujący sposób: 10+A1+D6, gdzie A1 to zawartość rejestru adresowego A1, D6 - zawartość rejestru danych D6 (jest to tryb adresowania pośredniego rejestrem adresowym z indeksem i przesunięciem - wszystkie tryby adresowania zostały opisane w Amigowcu 3-4/93). Jeśli testowany bit będzie wyzerowany, to znacznik Z zostanie ustawiony, w przeciwnym razie, znacznik Z zostanie wyzerowany.

Przykład: BTST D5,D0 powoduje sprawdzenie w rejestrze danych D0 bitu o numerze zawartym w rejestrze D5. Gdyby na przykład w D5 znajdowała się liczba 20, to przetestowany zostałby dwudziesty bit rejestru D0 i w zależności od jego stanu zostałby ustawiony znacznik Z.

Innym rozkazem z grupy operacji na bitach jest rozkaz BSET.

BSET - Test a bit and set. (Test bitu i ustawienie bitu.)

Zapis: BSET Dn,przeznaczenie lub BSET #n,przeznaczenie

Rozkaz BSET w pierwszej fazie testuje określony bit w operandzie przeznaczenia i w zależności od stanu tego bitu odpowiednio modyfikuje znacznik Z rejestru statusowego (robi to w taki sam sposób jak rozkaz BTST). Po dokonaniu testu dany bit jest ustawiany na jeden. Instrukcja BSET posiada identyczne ograniczenia co do rozmiaru w przypadku testowania zawartości pamięci jak BTST. Identyczny jest również sposób numeracji bitów.

Znaczniki:

X - Nie zmieniany.

N - Ustawiany, gdy testowany bit jest wyzerowany. W przeciwnym razie zerowany.

Z - Nie zmieniany.

V - Nie zmieniany.

C - Nie zmieniany.

Stosując rozkaz BSET nie możemy stosować wszystkich trybów adresowania dostępnych przy rozkazie BTST. Dozwolone są jedynie:

```
BSET ...,Dn)
BSET ...,(An)
BSET ...,(An)+
BSET ...,-(An)
BSET ...,p(An)
BSET ...,p(An,Rm.W)
BSET ...,p(An,Rm.L)
BSET ...,adr.W
BSET ...,adr.L
```

Przykład: BSET #7,$70000 powoduje sprawdzenie stanu 7 bitu w komórce pamięci o adresie $70000. Jeśli bit ten jest równy 1, to znacznik Z rejestru statusowego jest zerowany, gdy badany bit jest równy

ASM

0 - znacznik Z jest ustawiany na 1. Po odpowiednim ustawieniu znacznika Z rozkaz BSET ustawi 7 bit w komórce $70000 na 1.

Przykład: BSET #8,$70000 jest zapisem błędnym, gdyż przy operacjach na bitach w pamięci rozmiarem operacji jest bajt, czyli testować można bity o numerach od 0 do 7.

Rozkazem wykonującym operację odwrotną do wykonywanej przez BSET jest rozkaz BCLR.

BCLR - Test a bit and clear. (Test bitu i zerowanie bitu.)

Zapis: BCLR Dn,przeznaczenie lub BCLR #n,przeznaczenie

Rozkaz BCLR w pierwszym kroku testuje określony bit w operandzie przeznaczenia i w zależności od stanu tego bitu odpowiednio modyfikuje znacznik Z rejestru statusowego (robi to w taki sam sposób jak rozkazy BTST i BSET). Po dokonaniu testu dany bit zostaje wyzerowany. Instrukcja BCLR posiada takie same ograniczenia, co do rozmiaru w przypadku testowania zawartości pamięci, jak BTST i BSET. Identyczny jest również sposób numeracji bitów.

Znaczniki:

X - Nie zmieniany.

N - Ustawiany, gdy testowany bit jest wyzerowany. W przeciwnym razie zerowany.

Z - Nie zmieniany.

V - Nie zmieniany.

C - Nie zmieniany.

Przy używaniu instrukcji BCLR można stosować te same tryby adresowania co przy instrukcji BSET.

Przykład: BCLR D3,$70000 powoduje sprawdzenie w komórce pamięci o adresie $70000 stanu bitu o numerze zawartym w rejestrze danych D3 i odpowiednie ustawienie znacznika Z (zgodnie z kilkakrotnie już przedstawionymi zasadami). Po zmodyfikowaniu znacznika Z badany bit zostanie wyzerowany.

Ostatnim rozkazem z omawianej grupy jest rozkaz BCHG.

BCHG - Test a bit and change. (Test bitu i zamiana stanu bitu.)

Zapis: BCHG Dn,przeznaczenie lub BCHG #n,przeznaczenie

Rozkaz BCHG tak jak BSET i BCLR początkowo testuje określony bit w operandzie przeznaczenia i w zależności od jego stanu odpowiednio zmienia znacznik Z rejestru statusowego. Po wykonaniu testu rozkaz BCHG neguje zawartość badanego bitu. Operacja negacji oznacza, że jeśli badany bit był wyzerowany, to BCHG ustawi go na 1, a jeśli był ustawiony na 1, to BCHG go wyzeruje. Tak jak w przypadku BSET i BCLR instrukcja BCHG posiada rozmiar długiego słowa przy testowaniu bitów w rejestrach danych i rozmiar bajtu przy operacjach na komórkach pamięci.

Znaczniki:

X - Nie zmieniany.

N - Ustawiany, gdy testowany bit jest wyzerowany. W przeciwnym razie zerowany.

Z - Nie zmieniany.

V - Nie zmieniany.

C - Nie zmieniany.

Dopuszczalne tryby adresowania przy używaniu instrukcji BCHG są takie same jak przy BSET i BCLR.

Przykład: BCHG #5,D0 powoduje sprawdzenie zawartości piątego bitu rejestru danych D0 oraz odpowiednie ustawienie znacznika Z. Po uaktualnieniu stanu znacznika Z zawartość piątego bitu rejestru D0 zostanie zamieniona na przeciwną (0 na 1, a 1 na 0).

Można zauważyć, że gdyby nie istniał rozkaz BCHG pojedynczą instrukcję BCHG #5,D0 musielibyśmy zastąpić następującą sekwencją rozkazów:

```
    btst #5,D0
    beq Bit_Wyzerowany
Bit_Ustawiony:
    bclr#5,D0        ; wyzerowanie bitu jeśli był ustawiony
    bra Dalej
Bit_Wyzerowany:
    bset#5,D0        ; ustawienie bitu jeśli był wyzerowany
Dalej:
    ...
```

Spróbujemy teraz wykorzystać nowo zdobytą wiedzę w praktyce. Napiszemy programik, który po wciśnięciu lewego klawisza myszy mruga diodą POWER Amigi, natomiast po wciśnięciu prawego klawisza myszki program będzie kończył pracę.

Na początek zajmiemy się sprawdzaniem stanu lewego przycisku myszy. Za lewy klawisz myszki odpowiedzialny jest szósty bit rejestru o adresie $BFE001 (Piszę "rejestru", ale tak naprawdę jest to komórka pamięci tylko, że umieszczona nie w pamięci RAM czy ROM, lecz wewnątrz układów scalonych odpowiedzialnych za komunikację procesora z urządzeniami zewnętrznymi. W związku z tym "rejestr" można traktować jak zwykłą komórkę pamięci.) Otóż jeśli bit ten jest ustawiony na 1, oznacza to, że lewy przycisk myszy nie jest wciśnięty. Po wciśnięciu przycisku bit ten zostaje wyzerowany. Tak więc procedura czekająca na wciśnięcie lewego przycisku myszy powinna sprawdzać bit 6 pod adresem $BFE001 do momentu pojawienia się w nim 0:

```
;   ## Czekanie na wciśnięcie lewego klawisza myszki ##
Czekaj1:
    btst #6,$bfe001     ; sprawdzenie stanu 6 bitu w rejestrze $BFE001
    bne Czekaj1         ; jeśli równy 1 (przycisk nie wciśnięty),
                        ; to skok do etykiety Czekaj1
    rts                 ; wyjście z programu
```

Zrozumiałe chyba jest, że procedura czekająca na puszczenie lewego przycisku (oczywiście po wykryciu wciśnięcia tego przycisku) powinna czekać na pojawienie się na 6 bicie rejestru $BFE001 stanu 1. Fragment programu czekający na wciśnięcie i zwolnienie lewego przycisku powinien więc wyglądać tak:

```
;   ## Czekanie na wciśnięcie i zwolnienie lewego klawisza myszy ##
Czekaj1:
    btst #6,$bfe001     ; sprawdzenie stanu 6 bitu w rejestrze $BFE001
    bne Czekaj1         ; jeśli równy 1 (przycisk nie wciśnięty),
                        ; to skok do etykiety Czekaj1
Czekaj2:
    btst #6,$bfe001     ; sprawdzenie stanu 6 bitu w rejestrze $BFE001
    beq Czekaj2         ; jeśli równy 0 (przycisk nadal wciśnięty),
                        ; to skok do etykiety Czekaj2
    rts                 ; wyjście z programu
```

Wiemy już jak kontrolować lewy przycisk myszki. Natomiast o stanie prawego przycisku możemy dowiedzieć się testując stan bitu o numerze 2 w rejestrze o adresie $DFF016:

```
;   ## Czekanie na wciśnięcie prawego klawisza myszki ##
Czekaj3:
    btst #2,$dff016     ; sprawdzenie stanu 2 bitu w rejestrze $DFF016
    bne Czekaj3         ; jeśli równy 1 (prawy przycisk nie wciśnięty),
                        ; to skok do etykiety Czekaj3
    rts                 ; wyjście z programu
```

Jeśli potrafimy już wykryć wciśnięcie lewego lub prawego przycisku myszki, zajmijmy się mruganiem diodą POWER. Otóż aby przygasić diodę POWER (w starszych modelach A500 dioda nie tylko przygasa, ale nawet gaśnie całkowicie) należy ustawić na 1, pierwszy bit w rejestrze $BFE001 (pierwszy to znaczy ten o numerze 1). Natomiast wyzerowanie tego bitu sprawia, że dioda świeci pełną jasnością.

```
;   ## Przygaszenie diody POWER ##
ZgaśDiodę:
    bset#1,$bfe001
    rts

;   ## Zapalenie diody POWER ##
ZapalDiodę:
    bclr#1,$bfe001
    rts
```

W naszym programie będziemy zmieniali stan diody na przeciwny (z zapalonej na zgaszoną i ze zgaszonej na zapaloną) po wykryciu wciśnięcia lewego klawisza myszki. Po zmianie stanu diody będziemy czekali na puszczenie lewego przycisku, aby dioda nie mrugała jak oszalała. Natomiast w przypadku wciśnięcia prawego przycisku nasz program będzie kończył pracę:

```
;   ## Mruganie diodą ##
Czekaj1:
    btst #2,$dff016     ; sprawdzenie stanu prawego przycisku
    beq KoniecPracy     ; jeśli wciśnięty to skok do etykiety KoniecPracy
    btst #6,$bfe001     ; sprawdzenie stanu lewego przycisku
```

```
    bne Czekaj1        ; jeśli nie wciśnięty, to skok na początek programu
    bchg #1,$bfe001    ; zmiana stanu diody na przeciwny
Czekaj2:
    btst #6,$bfe001    ; sprawdzenie stanu lewego przycisku
    beq Czekaj2        ; jeśli nadal wciśnięty, to skok do etykiety Czekaj2
    bra Czekaj1        ; po zwolnieniu lewego przycisku
                       ; skok na początek programu
KoniecPracy:
    rts                ; wyjście z programu
```

W tym miejscu muszę zaznaczyć, że pokazany tutaj sposób kontrolowania przycisków myszy i zmiany stanu diody POWER nie należy do najbardziej eleganckich, gdyż odwołuje się bezpośrednio do rejestrów specjalizowanych układów scalonych. Na dotychczas produkowanych modelach Amig działa bez zarzutu, ale nie można zagwarantować, że Commodore nie wyprodukuje Amigi w której np. za stan przycisków myszki będzie odpowiedzialny zupełnie inny rejestr. Aby uniknąć przykrych niespodzianek w przyszłości należałoby testować przyciski myszy i mrugać diodą za pośrednictwem systemu operacyjnego. Jednak kurs ten ma być wprowadzeniem do programowania w języku asemblera, a nie szczegółowym opisem systemu operacyjnego.

Po opisie rozkazów z grupy operacji na pojedynczych bitach, przejdę do wspomnianych we wstępie instrukcji arytmetycznych.

Procesor Motorola 68000 posiada rozkazy wykonujące dodawanie, odejmowanie, mnożenie i dzielenie.

Rozkazy dodawania były już opisane w jednym z pierwszych odcinków tego kursu (w Amigowcu 5-6/93). Przypomnę je jedynie pokrótce:

ADD - dodawanie binarne (to po prostu zwykłe dodawanie).

ADDA - dodawanie adresu (tylko do dodawania do rejestrów adresowych).

ADDI - dodawanie natychmiastowe (do dodawania liczb podanych w sposób bezpośredni).

ADDQ - dodawanie szybkie (do dodawania liczb z zakresu od 1 do 8).

ADDX - dodawanie rozszerzone (dodawanie z uwzględnieniem znacznika rozszerzenia X).

Do odejmowania służy natomiast rozkaz SUB.

SUB - Substract binary. (Odejmowanie binarne.)

Zapis: SUB źródło,przeznaczenie

Rozkaz SUB odejmuje operand źródłowy od operandu przeznaczenia, a wynik zapisuje w miejscu przeznaczenia. Rozmiar operacji może być określony jako bajt, słowo lub długie słowo.

Znaczniki:

X - Ustawiany, gdy wystąpi pożyczka. W przeciwnym razie zerowany.

N - Ustawiany, gdy wynik jest ujemny. W przeciwnym razie zerowany.

Z - Ustawiany, gdy wynikiem operacji jest zero. W przeciwnym razie zerowany.

V - Ustawiany, gdy wystąpi nadmiar. W przeciwnym razie zerowany.

C - Ustawiany, gdy wystąpi pożyczka. W przeciwnym razie zerowany.

Dozwolone tryby adresowania dla operandu źródłowego:

SUB Dn, ...
SUB (An), ...
SUB (An)+, ...
SUB -(An), ...
SUB p(An), ...
SUB p(An,Dm.W), ...
SUB p(An,Dm.L), ...
SUB adr.W, ...
SUB adr.L, ...
SUB p(PC,Dn.W), ...
SUB p(PC,Dn.L), ...
SUB #nn, ...

Dozwolone tryby adresowania dla operandu docelowego:

SUB ... ,Dn
SUB ... ,(An)
SUB ... ,(An)+
SUB ... ,-(An)
SUB ... ,p(An)
SUB ... ,p(An,Dm.W)
SUB ... ,p(An,Dm.L)
SUB ... ,adr.W
SUB ... ,adr.L

Uwaga: Jeżeli w operandzie źródłowym użyliśmy innych trybów adresowania niż tryb adresowania bezpośredniego rejestru danych (SUB Dn, ...) lub tryb adresowania natychmiastowego (#nn), to w operandzie docelowym możemy użyć jedynie trybu adresowania bezpośredniego rejestru danych (SUB ... ,Dn).

Przykład: SUB.B (A0),(A1) jest zapisem nieprawidłowym i nie zostanie przyjęty przez program assemblera.

Przykład: SUB.W #34,D6 jest zapisem poprawnym i powoduje odjęcie liczby 34 od zawartości rejestru danych D6 (w rozmiarze słowa). Uzyskana różnica zostanie umieszczona w rejestrze D6.

Przykład: SUB.L (A1),D0 powoduje odjęcie długiego słowa z pamięci pod adresem zawartym w rejestrze adresowym A1 od rejestru danych D0. Różnica zostanie zapisana w rejestrze docelowym (w tym przypadku D0).

SUBA - Substract address. (Odejmowanie adresu.)

Zapis: SUBA ... ,An

Rozkaz SUBA powoduje odjęcie operandu źródłowego od rejestru adresowego określonego w polu przeznaczenia. Wynik jest zapisywany w rejestrze adresowym przeznaczenia. Rozmiar operacji może być określony jako słowo lub długie słowo, nie ma możliwości dodawania do rejestru adresowego danych w rozmiarze bajtu.

Znaczniki:

X - Nie zmieniany.
N - Nie zmieniany.
Z - Nie zmieniany.
V - Nie zmieniany.
C - Nie zmieniany.

Dozwolone tryby adresowania:

SUBA Dn,Am
SUBA An,Am
SUBA (An),Am
SUBA (An)+,Am
SUBA -(An),Am
SUBA p(An),Am
SUBA p(An,Dn.W),Am
SUBA p(An,Dn.L),Am
SUBA adr.W,An
SUBA adr.L,An
SUBA p(PC),An
SUBA p(PC,Dn.W),Am
SUBA p(PC,Dn.L),Am
SUBA #nn,An

Przykład: SUBA.L #2,A1 powoduje odjęcie dwójki od rejestru adresowego A1. Wynik zostanie zapisany w rejestrze A1.

SUBI - Substract immediate. (Odejmowanie natychmiastowe.)

Zapis: SUBI #nn, ...

Rozkaz SUBI odejmuje daną natychmiastową od operandu przeznaczenia, a wynik zapisuje w miejscu przeznaczenia. Rozmiarem operacji może być bajt, słowo lub długie słowo. Rozmiar danej natychmiastowej musi być zgodny z rozmiarem operacji. Oznacza to, że (przykładowo) rozkaz SUBI.B #$FFFF,D0 jest nieprawidłowy, gdyż liczba $FFFF nie daje się zapisać na jednym bajcie.

Znaczniki:

X - Ustawiany, gdy wystąpi pożyczka z najstarszego bitu. W przeciwnym wypadku zerowany.

N - Ustawiany, gdy wynik operacji jest ujemny. W przeciwnym razie zerowany.

Z - Ustawiany, gdy wynikiem operacji jest zero. W przeciwnym wypadku zerowany.

V - Ustawiany, gdy wystąpi nadmiar. W przeciwnym razie zerowany.

C - Ustawiany, gdy wystąpi pożyczka. W przeciwnym wypadku zerowany.

Dozwolone tryby adresowania:

SUBI #nn,Dn
SUBI #nn,(An)
SUBI #nn,(An)+
SUBI #nn,-(An)
SUBI #nn,p(An)
SUBI #nn,p(An,Dm.W)

SUBI #nn,p(An,Dm.L)
SUBI #nn,adr.W
SUBI #nn,adr.L

Przykład: SUBI.B #10,-(A3) powoduje zmniejszenie zawartości rejestru adresowego A3 o jeden (tryb adresowania z predekrementacją), a następnie odjęcie liczby 10 od zawartości komórki pamięci o adresie zawartym w rejestrze A3. Wynik zostaje zapisany w tej komórce.

SUBQ - Substract quick. (Odejmowanie szybkie.)

Zapis: SUBQ #n, ...

Rozkaz SUBQ odejmuje daną natychmiastową od operandu przeznaczenia, a wynik zapisuje w miejscu przeznaczenia. Różnicą między tą instrukcją a instrukcją SUBI jest fakt, że w przypadku SUBQ dana natychmiastowa może zawierać się jedynie w zakresie od 1 do 8. Rozmiar operacji może być określony jako bajt, słowo lub długie słowo. Operacje o długości słowa lub długiego słowa dozwolone są również na rejestrach adresowych.

Znaczniki:

X - Ustawiany, gdy wystąpi pożyczka. W przeciwnym razie zerowany.

N - Ustawiany, gdy wynik operacji jest ujemny. W przeciwnym razie zerowany.

Z - Ustawiany, gdy wynikiem operacji jest zero. W przeciwnym razie zerowany.

V - Ustawiany, gdy wystąpi nadmiar. W przeciwnym razie zerowany.

C - Ustawiany, gdy wystąpi pożyczka. W przeciwnym razie zerowany.

W przypadku operacji na rejestrze adresowym nie są zmieniane żadne znaczniki rejestru statusowego.

Dozwolone tryby adresowania:

SUBQ #n,Dn
SUBQ #n,An
SUBQ #n,(An)
SUBQ #n,(An)+
SUBQ #n,-(An)
SUBQ #n,p(An)
SUBQ #n,p(An,Dm.W)
SUBQ #n,p(An,Dm.L)
SUBQ #n,adr.W
SUBQ #n,adr.L

Przykład: SUBQ.B #2,A0 jest zapisem nieprawidłowym, gdyż od rejestrów adresowych nie wolno odejmować danych w rozmiarze bajtu.

Przykład: SUBQ.L #2,A0 jest już zapisem najzupełniej poprawnym i spowoduje odjęcie dwójki od rejestru adresowego A0.

Przykład: SUBQ.L #10,A0 jest znowu zapisem błędnym, gdyż instrukcja SUBQ umożliwia odejmowanie liczb z zakresu 1 do 8 (10 leży poza tym zakresem).

SUBX - Substract extended. (Odejmowanie z rozszerzeniem.)

Zapis: SUBX Dn,Dm lub SUBX -(An),-(Am)

Rozkaz SUBX odejmuje operand źródłowy od operandu przeznaczenia wraz z bitem rozszerzenia, a wynik zapisuje w miejscu przeznaczenia. Rozmiar operacji może być określony jako bajt, słowo lub długie słowo.

Znaczniki:

X - Ustawiany, gdy wystąpi pożyczka. W przeciwnym razie zerowany.

N - Ustawiany, gdy wynik operacji jest ujemny. W przeciwnym razie zerowany.

Z - Zerowany, gdy wynik operacji jest różny od zera. W przeciwnym razie nie zmieniany.

V - Ustawiany, gdy wystąpi nadmiar. W przeciwnym razie zerowany.

C - Ustawiany, gdy wystąpi pożyczka z najstarszego bitu. W przeciwnym razie zerowany.

Przykład: SUBX.L D0,D1 powoduje odjęcie zawartości rejestru D0 od zawartości rejestru D1. Wynik (w rozmiarze długiego słowa) zostaje zapisany w rejestrze D1. Oprócz tego, jeżeli znacznik rozszerzenia X był ustawiony, zawartość rejestru D1 zostaje dodatkowo zmniejszona o jeden.

Z rozkazów dodawania i odejmowania pozostały już tylko rozkazy ABCD i SBCD. Służą one do dodawania i odejmowania liczb zapisanych w kodzie BCD.

Kod BCD opiera się na kodzie szesnastkowym z tym, że nie są używane kombinacje od 10 do 15 (to znaczy od $A do $F). Każda czwórka bitów reprezentuje jedną cyfrę dziesiętną, tak więc w jednym bajcie przechowywane są dwie cyfry dziesiętne. Aby lepiej zrozumieć kodowanie w kodzie BCD przedstawię pierwsze 16 liczb w zwykłym kodzie dwójkowym i w kodzie BCD:

	kod dwójkowy	kod BCD
0	00000000	00000000
1	00000001	00000001
2	00000010	00000010
3	00000011	00000011
4	00000100	00000100
5	00000101	00000101
6	00000110	00000110
7	00000111	00000111
8	00001000	00001000
9	00001001	00001001
10	00001010	00010000
11	00001011	00010001
12	00001100	00010010
13	00001101	00010011
14	00001110	00010100
15	00001111	00010101

Kod BCD jest szczególnie przydatny przy dodawaniu i odejmowaniu liczb zapisanych w formacie znakowym (czyli takim, jakiego zawsze używamy wpisując zwykłe liczby z klawiatury). Chcąc zamieniać dane liczby na postać binarną po to, żeby dokonać obliczeń, a następnie ponownie zamieniać je na format znakowy, należałoby użyć instrukcji mnożenia i dzielenia, które są instrukcjami stosunkowo wolnymi. Użycie kodu BCD eliminuje te czasochłonne działania umożliwiając szybką zamianę na i z formatu znakowego.

ABCD - Add decimal with extend. (Dodawanie dziesiętne z rozszerzeniem.)

Zapis: ABCD Dn,Dm lub ABCD -(An),-(Am)

Instrukcja ABCD dodaje operand źródłowy do operandu przeznaczenia wraz z bitem rozszerzenia X. Wynik jest zapisywany w miejscu przeznaczenia. Dodawanie jest przeprowadzane z wykorzystaniem kodowania BCD. Rozkaz ABCD posiada jedynie rozmiar bajtu.

Znaczniki:

X - Ustawiany, gdy wystąpiło przeniesienie dziesiętne. W przeciwnym wypadku zerowany.

N - Przyjmuje wartości nieokreślone.

Z - Zerowany, gdy wynik jest różny od zera. W przeciwnym razie nie zmieniany.

V - Przyjmuje wartości nieokreślone.

C - Ustawiony, gdy wystąpiło przeniesienie dziesiętne. W przeciwnym razie zerowany.

Przykład: ABCD.B D0,D1 powoduje dodanie zawartości rejestru D0 do zawartości rejestru D1. Wynik zostaje zapisany w rejestrze D1. Jeśli przed operacją znacznik X był ustawiony, to dodatkowo rejestr D1 zostaje zwiększony o 1. Liczby w D0 i D1 są traktowane jak zapisane w kodzie BCD.

SBCD - Substract decimal with extend. (Odejmowanie dziesiętne z rozszerzeniem.)

Zapis: SBCD Dn,Dm lub SBCD -(An),-(Am)

Rozkaz SBCD odejmuje od operandu przeznaczenia operand źródłowy oraz bit rozszerzenia X. Wynik odejmowania jest zapisywany w miejscu przeznaczenia. Rozmiar operacji SBCD jest ograniczony do bajtu. Odejmowane liczby są traktowane jako zapisane w kodzie BCD.

Znaczniki:

X - Ustawiany, gdy wygenerowana została pożyczka w najbardziej znaczącej cyfrze BCD. W przeciwnym razie zerowany.

N - Przyjmuje wartości nieokreślone.

Z - Zerowany, gdy wynik odejmowania jest różny od zera. W przeciwnym wypadku nie zmieniany.

V - Przyjmuje wartości nieokreślone.

C - Ustawiany, gdy wystąpiła pożyczka w najbardziej znaczącej cyfrze BCD. W przeciwnym razie zerowany.

Przykład: SBCD.B D4,D2 odejmuje od zawartości rejestru danych

D2 zawartość rejestru D4. Wynik odejmowania jest zapisywany w rejestrze D2. Jeżeli przed operacją był ustawiony znacznik X , to dodatkowo rejestr D2 zostaje zmniejszony o jeden. Liczby w rejestrach są traktowane jak zapisane w kodzie BCD.

Z instrukcji arytmetycznych wykonywanych przez procesor Motorola 68000 pozostały już tylko instrukcje mnożenia i dzielenia.

MULU - Unsigned multiply. (Mnożenie bez znaku.)

Zapis: MULU źródło,Dn

Rozkaz MULU mnoży 16-bitową liczbę określoną przez operand źródłowy przez 16-bitową liczbę zawartą w młodszym słowie określonego rejestru danych. 32-bitowy wynik zapisywany jest w rejestrze danych. Instrukcja działa w arytmetyce bezznakowej (najstarszy bit mnożonych liczb nie jest traktowany jako bit znaku).

Znaczniki:

X - Nie zmieniany.

N - Ustawiany, gdy najbardziej znaczący bit wyniku jest równy jeden. W przeciwnym wypadku zerowany.

Z - Ustawiany, gdy wynikiem mnożenia jest zero. W przeciwnym razie zerowany.

V - Zawsze zerowany.

C - Zawsze zerowany.

Przy rozkazie MULU dozwolone są następujące tryby adresowania:

MULU Dm,Dm
MULU (An),Dm
MULU (An)+,Dm
MULU -(An),Dm
MULU p(An),Dm
MULU p(An,Ri.W),Dm
MULU p(An,Ri.L),Dm
MULU adr.W,Dm
MULU adr.L,Dm
MULU p(PC),Dm
MULU p(PC,Rn.W),Dm
MULU p(PC,Rn.L),Dm
MULU #nn,Dm

Przykład: MULU #100,D0 powoduje pomnożenie zawartości rejestru danych D0 przez 100. Wynik mnożenia zostaje zapisany w rejestrze D0.

MULS - Signed multiply. (Mnożenie ze znakiem.)

Zapis: MULS źródło,Dn

Instrukcja MULS mnoży 16-bitową liczbę określoną operandem źródłowym przez 16-bitową liczbę zawartą w młodszym słowie określonego rejestru danych. 32-bitowy wynik jest zapisywany w rejestrze danych. Instrukcja wykonuje operacje na liczbach zapisanych w kodzie uzupełnień do dwóch (kod U2 został opisany w pierwszym odcinku tego kursu - Amigowiec 3-4/93).

Znaczniki:

X - Nie zmieniany.

N - Ustawiany, gdy wynik jest ujemny. W przeciwnym razie zerowany.

Z - Ustawiany, gdy wynikiem mnożenia jest zero. W przeciwnym razie zerowany.

V - Zawsze zerowany.

C - Zawsze zerowany.

Przy instrukcji MULS można stosować takie same tryby adresowania jak przy instrukcji MULU.

Przykład: MULS #-1,D0 powoduje pomnożenie liczby zapisanej na młodszym słowie rejestru danych D0 przez -1. Mnożenie przez -1 można wykorzystać do zmiany znaku liczby (chociaż nie jest to najszybszy z możliwych sposobów).

DIVU - Unsigned divide. (Dzielenie bez znaku.)

Zapis: DIVU źródło,Dn

Instrukcja DIVU dzieli 32-bitową liczbę zawartą w rejestrze danych przez 16-bitową liczbę określoną przez operand źródłowy. W wyniku dzielenia młodsze słowo rejestru danych zawiera część całkowitą ilorazu, natomiast starsze słowo - resztę z dzielenia. Wszystkie liczby traktowane są jako liczby bez znaku.

Podczas wykonywania rozkazu DIVU mogą wystąpić dwa błędy:

- Dzielenie przez zero. W tym wypadku procesor Motorola 68000 przechodzi do obsługi stanu wyjątkowego (Jeżeli nie mamy na Amidze uruchomionego programu przechwytującego błąd dzielenia przez zero oznacza to Guru).

- Dzielenie dużej liczby przez liczbę małą. Część całkowita ilorazu nie mieści się na szesnastu bitach. Zostaje wtedy ustawiony bit V rejestru statusowego, a zawartość rejestru danych nie zostaje zmieniona.

Znaczniki:

X - Nie zmieniany.

N - Ustawiany, gdy iloraz jest ujemny. W przeciwnym wypadku zerowany. W przypadku wystąpienia nadmiaru przyjmuje wartości nieokreślone.

Z - Ustawiany, gdy wynik dzielenia jest zerowy. W przeciwnym razie zerowany. W przypadku wystąpienia nadmiaru przyjmuje wartości nieokreślone.

V - Ustawiany, gdy wystąpił nadmiar dzielenia. W przeciwnym razie zerowany.

C - Zawsze zerowany.

Dozwolone tryby adresowania przy rozkazie DIVU to:

DIVU Dn,Dm
DIVU (An),Dm
DIVU (An)+,Dm
DIVU -(An),Dm
DIVU p(An),Dm
DIVU p(An,Ri.W),Dm
DIVU p(An,Ri.L),Dm
DIVU adr.W,Dm
DIVU adr.L,Dm
DIVU p(PC),Dm
DIVU p(PC,Rn.W),Dm
DIVU p(PC,Rn.L),Dm
DIVU #nn,Dm

Przykład: DIVU #10,D1 powoduje podzielenie zawartości rejestru danych D1 przez 10. Załóżmy, że rejestr D1 zawiera liczbę 105. Po wykonaniu operacji DIVU #10,D1 w młodszym słowie rejestru D1 będzie znajdowała się liczba 10 (część całkowita z dzielenia 105/10=10), natomiast starsze słowo rejestru D1 będzie zawierało liczbę 5 (reszta z dzielenia 105/10=5).

DIVS - Signed divide. (Dzielenie ze znakiem.)

Zapis: DIVS źródło,Dn

Instrukcja DIVS dzieli 32-bitową liczbę zapisaną w rejestrze danych przez 16-bitową liczbę określoną przez operand źródłowy. W wyniku tego dzielenia młodsze słowo rejestru danych zawiera część całkowitą ilorazu, natomiast starsze słowo rejestru - resztę z dzielenia (tak samo jak w przypadku DIVU). Iloraz ma znak zgodny z ogólnie znanymi z zasadami matematycznymi, natomiast reszta ma zawsze znak liczby zawartej w rejestrze danych przed operacją dzielenia. Dzielone liczby są traktowane jak zapisane w kodzie uzupełnień do dwóch.

Przy wykonywaniu instrukcji DIVS mogą wystąpić takie same błędy jak przy DIVU. W taki sam również sposób zachowuje się procesor w razie ich wystąpienia.

Znaczniki:

X - Nie zmieniany.

N - Ustawiany, gdy iloraz jest ujemny. W przeciwnym razie zerowany. W przypadku wystąpienia nadmiaru przyjmuje wartości nieokreślone.

Z - Ustawiany, gdy wynik dzielenia jest równy zero. W przeciwnym wypadku zerowany. W przypadku wystąpienia nadmiaru przyjmuje wartości nieokreślone.

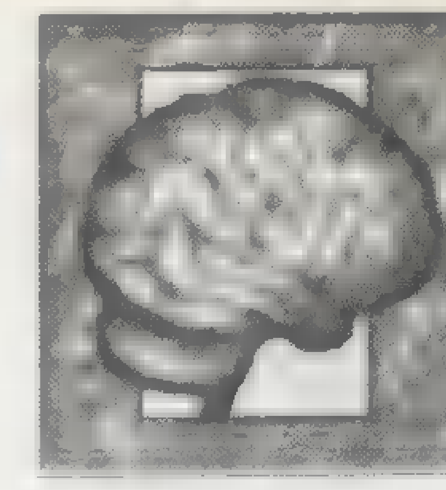
V - Ustawiany, gdy wystąpił nadmiar dzielenia. W przeciwnym razie zerowany.

C - Zawsze zerowany.

Z instrukcją DIVS można stosować te same tryby adresowania jak przy instrukcji DIVU.

Przykład: DIVS #-10,D0 powoduje podzielenie liczby zawartej w rejestrze danych D0 przez -10. Załóżmy, że rejestr D0 zawiera liczbę 205, to po wykonaniu operacji DIVS #-10,D0 w młodszym słowie rejestru D0 będzie znajdowała się liczba -20 (część całkowita z dzielenia 205/-10=-20), natomiast starsze słowo rejestru D0 będzie zawierało liczbę +5 (reszta z dzielenia 205/10=5). Gdyby natomiast rejestr D0 zawierał liczbę -205, to po dzieleniu młodsze słowo rejestru danych zawierałoby liczbę 20 (część całkowita z dzielenia -205/-10=20), a starsze słowo liczbę -5 (reszta z dzielenia -205/-10=5 ze znakiem liczby w rejestrze D0 przed dzieleniem (-205), czyli -5).

Poznaliśmy więc już wszystkie rozkazy arytmetyczne wykonywane przez procesor Motorola 68000. W następnym Amigowcu rozpoczniemy pisanie programu prostego kalkulatora wykonującego podstawowe operacje matematyczne. ❑



Na początku chcielibyśmy podziękować za listy z rozwiązaniami krzyżówki zamieszczonej w numerze świątecznym (9-10/93). Chwała Wam, gdyż przypadkowo wydrukowaliśmy tylko część krzyżówki. Rozwiązaniem jest hasło, które nawiązuje do zeszłorocznego przeboju kinowego Stevena Spielberga pt. "Park Jurajski":

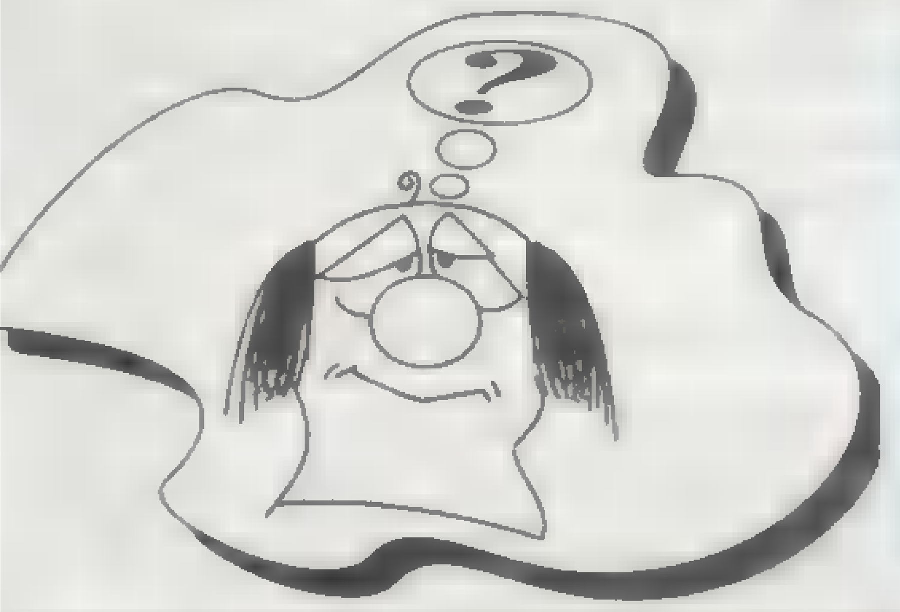
"Dzięki technice komputerowej spotkały się dinozaury, które dzieliło sto milionów lat"

Nagrody wylosowali:
Mirosław Stelmaszczak - Nowa Cerkiew
Zygmunt Zberek - Lubiniec
Wojciech Andryszczak - Lubliniec

Zapraszam na kolejną porcję zadań. Sześć łamigłówek zostało podzielonych na trzy grupy. Każda grupa wyróżnia się z pozostałych odpowiednim obrazkiem. Żeby wziąć udział w losowaniu należy poprawnie rozwiązać przynajmniej jedno zadanie z każdej grupy, ale nie mniej niż cztery zadania z całości. Kupony z rozwiązaniami prosimy przesyłać do 15 kwietnia pod adresem:

Redakcja "Amigowca",
dział Rusz Głową,
ul. Świętojańska 2/7,
85-017 Bydgoszcz.

Rozwiązanie w numerze 4/93. Kupon konkursowy znajduje się na stronie 66.



Zadanie 1

Poniżej podany jest ciąg liter. Jaką literę należy wstawić w miejsce znaku zapytania ?

D P N N P E P S ?

Zadanie 2

W jaki sposób można zapisać liczbę dziewięćdziesiąt przy pomocy trzech "5". Możliwe jest wstawianie pomiędzy cyframi tylko znaków matematycznych. Uwaga! Na to pytanie powinien odpowiedzieć każdy szanujący się komputerowiec i programista.

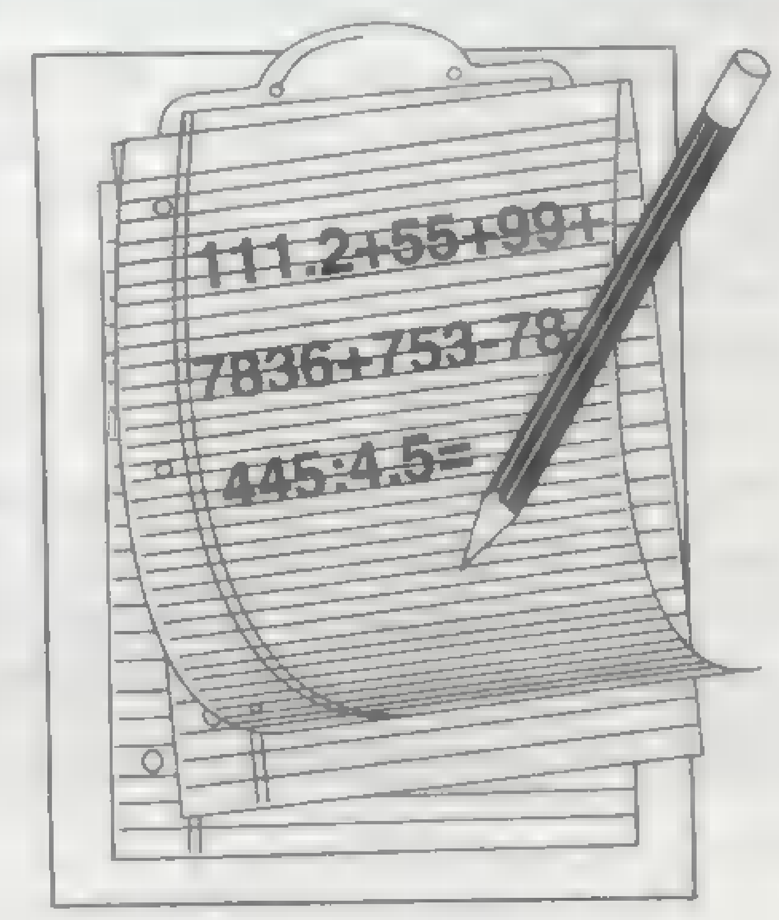
Zadanie 3

W jednym z wagonów pociągu jadącego ze stacji A do stacji B z prędkością 72 km/h znajduje się komar. Komar lata wzdłuż wagonu od jednego końca do drugiego z prędkością 108 km/h (względem wagonu). Wagon ma długość 30m. Jaką drogę (względem ziemi) przeleci komar, jeżeli stacje A i ■ oddalone są od siebie o 36 km?

Zadanie 4

Komputerlandia to taki dziwny kraj. Zamieszkujący go Bitowcy żywią się akcesoriami komputerowymi. Szczególnym przysmakiem są dyskietki 3,5", które sprzedaje się w pudełkach po 6, 9 lub 20 sztuk. Jeżeli Bitowiec chce kupić 15 dyskietek, to bierze paczkę z 6 i 9 sztukami. Jeżeli tego dnia jego rodzina ma ochotę na 21 dyskietek, kupuje dwie małe paczki i jedną średnią. Problem powstaje w momencie, gdy Bitowiec ma ochotę na akurat np. 11 lub 16 dyskietek, ponieważ żadna kombinacja 6, 9 i 20 nie daje w sumie 11 i 16. Niestety sprzedawcy nie pozwalają na otwieranie pudełek i dobieranie pojedynczych sztuk, ponieważ dyskietki bardzo szybko dostają wirusa po rozerwaniu hermetycznego opakowania.

Jaka jest zatem największa liczba dyskietek, której Bitowiec nie może kupić? Należy pamiętać o tym, że Bitowcy to oszczędny szczep i nigdy nie kupują za dużo dyskietek (za mało też nie bo są bardzo chytrzy).



Zadanie 5

Poniżej podana jest myśl Honore de Balzaca w zaszyfrowanej formie. Zasady według których odbyło się szyfrowanie są następujące:

a. po pierwsze wszystkie znaki w pierwotnym tekście zamieniono na odpowiednie liczby zgodnie z kodem ASCII,
b. od każdej w ten sposób powstałej liczby odjęto 163 i otrzymany wynik pomnożono przez -1 (minus jeden),
c. uzyskany ciąg pogrupowano po trzy liczby. Pierwszą liczbę w każdej grupie nazwano a, drugą b, i trzecią c. Wykonano następujące działania: d=a+b-c+64, e=a-b+c+37, f=(a+b+c)*2. Zmienne d, e, f z każdej trójki utworzyły ciąg, który jest przedstawiony poniżej:

147,130,374; -56,-93,-4; 315,-86,2; 296,-113,-36; 104,85,328; 306,-95,-16; 135,92,318; 265,-48,78; -251,210,-154; -66,-83,-28; -62,287,0; 108,97,296; 110,117,324; -66,-83,-28; 107,122,334; 59,162,458.

Zadanie polega na odszyfrowaniu tej myśli (postępowanie odwrone do przedstawionego wyżej).

Przykład szyfrowania: Weźmy następujący tekst: "Raj" i zamieńmy go na kod ASCII a otrzymamy: 82, 97, 106. Po wykonaniu działań z punktu (b) otrzymujemy: 81, 66, 57. W tym przypadku powstała tylko jedna trzyliczbowa grupa dla której a=81, b=66 i c=57. Zgodnie z punktem (c) wyliczamy, że d=154, e=109, f=408, zatem zaszyfrowany tekst ma postać: 154, 109, 408;



Zadanie ■

Liczba pierwsza p jest to taka liczba naturalna większa od jedności, której jedynymi dzielnikami są 1 oraz p. Kolejnymi liczbami pierwszymi są: 2,3,5,7,11,13,17,19,... Istnieje nieskończenie wiele liczb pierwszych. Jedną z wielu metod ich wyznaczania jest tzw. sito Eratostenesa. Metoda ta polega na sporządzeniu tablicy liczb naturalnych większych od 1. Pierwsza liczba w tablicy to 2. Wykreślamy z tablicy wszystkie liczby podzielne przez 2 (czyli 4,6,8,10...). Następną nieskreśloną liczbą jest 3. Należy zatem wykreślić wszystkie liczby, które są przez nią podzielne (6,9,12,...). Kolejna nieskreślona liczba po 2 i 3 to 5. Podobnie jak poprzednio, po wyeliminowaniu wszystkich liczb, których jest ona dzielnikiem, jako kolejną nieskreśloną liczbę w tablicy znajdujemy 7, itd. Wykonując te działania, dojdziemy wreszcie do tego, że wszystkie niewykreślone liczby w tablicy to liczby pierwsze.

Zadanie polega na tym, żeby przy pomocy metody sita Eratostenesa (albo dowolnej innej metody) znaleźć liczbę pierwszą jak najbardziej zbliżoną do pięciu milionów. Zadanie to można wykonać nawet na Amidze z 1MB pamięci!

Kupując A600 kierowałem się wbudowanym wewnętrznym kontrolerem AT-Bus. Obecnie jestem zainteresowany dokupieniem dysku twardego... Dowiedziałem się, że nie mogę tego zrobić w wersji Kickstartu 37.299. Proszę o ustosunkowanie się do tej informacji.

Bardzo zaskoczyło mnie to pytanie. Nigdy jeszcze nie spotkałem się z tym, aby możliwość zainstalowania dysku twardego była związana z wersją Kickstartu. Sądzę, że musiała tu zajść jakaś pomyłka i proponuję zbytnio się nie przejmować tego typu informacjami.

Czy gry i programy na A1200 będą chodzić po podłączeniu komputera do telewizora przez złącze SCART?

Podłączenie Amigi do telewizora przez popularne złącze "EURO-SCART" nie wpływa w żaden ujemny sposób na działanie komputera. Oznacza to, że wszystkie programy i gry normalnie chodzące na A1200 będą działały bez zarzutu. Przed zakupem jakichkolwiek nowych programów proponuję jednak upewnić się, czy współpracują one z Kickstartem 3.0 zainstalowanym standardowo w A1200. Jeżeli okaże się, że nasza ulubiona gra nie chce się "odpalić", wówczas mamy do wyboru trzy możliwości. Pierwsza, zmienić ulubioną grę na inną (to bardzo ciężka decyzja). Druga, zaopatrzyć się w przełącznik Kickstartów 1.3/3.0 (działający podobnie jak znany z A500 przełącznik 1.3/2.0). Trzecia, skorzystać z programowej emulacji systemu 1.3

Warto jeszcze wspomnieć, że złącze "EURO-SCART" przyjmuje sygnał RGB wysyłany z komputera, przez co następuje wyraźna poprawa jakości obrazu w porównaniu do tego, jaki uzyskujemy stosując modulator telewizyjny.

Czy na A1200 z monitorem VGA lub SVGA będą chodzić wszystkie gry i programy?

Niestety nie. W przypadku programów użytkowych problem jest niewielki, gdyż większość z nich korzysta z driverów systemowych i uruchamia się z ich wykorzystaniem w dowolnie wybranym przez użytkownika trybie wyświetlania. Osobiście korzystam z takiego właśnie zestawu (A1200 + monitor SVGA) i napotkałem zaledwie kilka programów, które nie chciały pracować z monitorem VGA (uruchamiały się na własnych (custom), niestandardowych ekranach, np.: "Digi Paint 3", "The Disney Animation Studio", "Lapunovia"). W wielu przypadkach bardzo pomocnym okazał się niewielki programik "ForceVGA", który powoduje, że programy uruchamiają się w trybie 30 Khz możliwym do wyświetlenia przez zwykły monitor VGA (zbliżoną funcję spełnia program "AmigaToDoubleNTSC", przy czym w jego przypadku wszystkie programy będą się uruchamiały w trybie "DoubleNTSC" pracy monitora). Zagadnienia dotyczące trybów pracy monitorów znalazły się w poprzednim wydaniu "Pism, pisemek".

Jeżeli chodzi o gry, to sytuacja nie jest zbyt wesoła. Z powodu braku możliwości uruchomienia jakiegokolwiek programu pomocniczego automatycznie odpadają wszystkie gry typu "Not a DOS" (czyli praktycznie rzecz biorąc grubo ponad połowa, a może i więcej). W większości pozostałych przypadków sytuacja niestety jest podobna, gdyż pomimo tego, że niektóre gry uruchamiają się z Workbencha, to jednak w następnej kolejności otwierają własny ekran i zabawa zostaje radykalnie przerwana (np. w grze "Gunship 2000" można spokojnie obejrzeć czołówkę, dokonać wyboru pilota i misji, uzbroić śmigłowiec i... koniec, gdyż w momencie, w którym na ekran "wchodzi wektorówka", zostaje zerwana częstotliwość pracy monitora i ekran "wariuje"). W tym przypadku należy jednak próbować wykorzystać programy, o których wspomniałem powyżej. Istnieje jednak kilka gier, przeważnie starszej już daty, które pomimo tych wszystkich trudności potrafią się uruchomić i prawidłowo działać (np.: "Sargon III", "Dune II", "Sim City II"). Warto także zauważyć, że zaczynają pojawiać się w większej ilości gry, które umożliwiają samodzielne wybranie trybu pracy.

Podsumowując. Monitor VGA z A1200 nadaje się raczej do profesjonalnej pracy i odradzałbym tworzenie takiego zestawu osobom niedoświadczonym.

Gdzie można dostać karty PCMCIA z dodatkową pamięcią RAM i ile one kosztują?

Czy można odczytać na A600 karty PCMCIA z IBM i czy potrzebny jest do odczytu jakiś specjalny program?

Te dwa pytania postanowiłem połączyć razem, gdyż dotyczą tego samego zagadnienia. Posiadacze A600 i A1200 oglądając swój komputer nie mogli przegapić wąskiej szczeliny znajdującej się z lewej strony ich komputera. Tam właśnie znajduje się owo tajemnicze złącze: PCMCIA. Podobno służy ono do rozbudowy komputera, dlaczego więc go nie wykorzystać? I właśnie w tym miejscu zaczynają się problemy. Gdzie zakupić tego typu karty? Po części z obowiązku, a po części z ciekawości przewertowałem sterty literatury komputerowej ("PC-kurier", "Enter", "Magazyn Amiga", "C&A" itp) w poszukiwaniu firm zajmujących się dystrybucją tego typu rozszerzeń. Spotkałem się tylko z dwiema propozycjmi obejmującymi karty PCMCIA z 2 lub 4 MB 16-bitowej pamięci RAM (dla zainteresowanych podaję numery telefonów: (22) 26-10-10 Warszawa "Computer Project", (066) 326-714 Września firma "Eureka"). Trudno powiedzieć, czy inne przedsiębiorstwa nie są zainteresowane reklamowaniem takiego "drobiazgu", czy też nie posiadają go w swoich ofertach. Z drugiej strony podawanie cen z zagranicznych pism (szczególnie angielskich) mija się z celem, gdyż są to zazwyczaj wartości zupełnie nie przeliczalne na polskie warunki.

Pozostaje mi jeszcze podać kilka niezwykle ważnych uwag dotyczących drugiego pytania. Otóż w A600 i A1200 można bezproblemowo stosować karty PCMCIA z rozszerzeniami pamięci pochodzącymi od IBMa. Dzieje się tak dlatego, że znajdujące się w Amigach złącze PCMCIA-I jest standardowe dla wszystkich rodzajów komputerów (na takiej samej zasadzie jak: AT-Bus, SCSI, itp.) i nie potrzeba żadnych specjalnych programów do obsługi tego typu rozszerzeń. Uwaga! Złącze PCMCIA-I jest 16-bitowe, podczas gdy standardem w chwili obecnej jest 32-bitowa PCMCIA-2 (a coraz częściej 3). Przy ewentualnym zakupie należy zwrócić na to szczególną uwagę. Druga sprawa dotyczy A1200. Mija się bowiem z celem stosowanie 16-bitowych rozszerzeń pamięci w 32-bitowym komputerze (w końcu tego typu rozszerzanie możliwości komputera ma przyspieszać jego pracę, a nie zwalniać). Na pocieszenie pozostaje fakt, że PCMCIA znajdująca się w Amigach przeżywa właśnie swoją drugą młodość za sprawą dysku twardego dołączanego przez to złącze. Być może spowoduje to większe zainteresowanie polskich dystrybutorów tym zagadnieniem.

Czy możliwa jest wymiana procesora Motorola 68000/7,14 MHz na mikroprocesor 68EC020/14,19 MHz w A500 i co by to ewentualnie mogło dać?

Bezpośrednia wymiana tego typu nie jest możliwa. Różnice pomiędzy procesorami 68000 i 68020 (pomimo zgodności programowej) są zbyt duże. Również architektura komputerów, w których są one stosowane diametralnie się różni (16-bitowa w A500, 32-bitowa w A1200). Tak więc, nawet gdyby udało się wstawić MC 68020 w miejsce starego procesora, to poprawa szybkości byłaby niewielka (szybki procesor zmuszony byłby do pracy z powolną 16-bitową pamięcią). Oczywiście poradzono sobie i z tym problemem. Powstały bowiem specjalne karty i rozszerzenia "dopalające" A500. Ogólnie tego typu urządzenia dzielą się na wewnętrzne

i zewnętrzne.

Instalacja dopalaczy wewnętrznych polega na otwarciu komputera (co wiąże się z utratą gwarancji), wyjęciu procesora 68000 i wstawieniu w jego miejsce odpowiedniej płytki. Znajduje się na niej nie tylko nowy procesor, ale również cała "otoczka" niezbędna do jego prawidłowej pracy (m. in. dodatkowa pamięć RAM, miejsce na koprocesor, itd). W ten sposób można zaopatrzyć się nie tylko w procesor 68020, ale również 68030, co zapewnia oczywiście znacznie większe osiągi poczciwej A500. Wiele firm oferuje również karty turbo z procesorem 68000 o podwyższonej częstotliwości zegara do 14.28 MHz (a nawet 28 MHz!). Jest to rozwiązanie stosunkowo tanie, jednak należy wziąć pod uwagę, że nawet najszybszy 68000 (czy 68020) nie dogoni procesora 68030 taktowanego zegarem 16 MHz (czyli najwolniejszym z możliwych).

Karty zewnętrzne są podłączane przez złącze "Expansion Port". W zgrabnych obudowach, dopasowanych kształtem do komputera, można znaleźć praktycznie wszystko o czym zamarzy dusza posiadacza Amigi. Począwszy od dowolnego procesora i koprocesora, przez 32-bitową pamięć RAM, po kontroler SCSI wraz z dyskiem twardym. W odpowiedniej konfiguracji nasza poczciwa A500 może pobić nawet A3000. Jednak nie ma róży bez kolców. Takie "przyspieszacze" są jednak niezwykle drogie. Dla przykładu, oficjalne przedstawicielstwo firmy GVP proponowało (stan z grudnia 93): rozszerzenie "A520-Turbo+" (procesor 68EC030-40MHz, możliwość zainstalowania do 8 MB FAST RAM, miejsce na koprocesor 68882 40MHz, złącze do karty emulującej IBM PC) + 1 MB 32-bit RAM + dysk twardy 80 MB (SCSI) za 13,6 mln. zł.

Gdzie można kupić rozszerzenia pamięci do A1200 i ile one kosztują?

W chwili obecnej rozszerzenia pamięci do A1200 pojawiły się w ofertach prawie wszystkich firm komputerowych, zajmujących się hardwarem do Amigi. Sprzyja temu rosnąca w Polsce popularność A1200. Jednym z efektów tego zainteresowania jest pojawienie się w pełni rodzimego rozszerzenia pamięci o symbolu E1204. Dystrybutorem tego urządzenia jest firma "Elsat" (tel. (22) 40-58-76 lub (2) 642-96-05). Pozwala ono na rozszerzenie pamięci od 1 do 8 MB (FAST RAM) oraz na zainstalowanie koprocesora (68881/2). Cena wyjściowa "łysej" karty (tzn. 0 RAM i bez koprocesora) wynosi 2,99 mln zł. Z inną propozycją wystąpiła firma "Eureka" (ta sama, która oferowała karty PCMCIA) sprzedając "Memory Master 1200" z 1MB RAM za 3,4 mln zł (dodatkowe 2 MB to koszt 1,68 mln. zł). Nie sposób umieścić tutaj wszystkich ofert, z resztą dział "Pism, pisemek" nie powinien spełniać funkcji reklamowych i dlatego proponuję czytelnikom samodzielną penetrację szeregu ogłoszeń ukazujących się w czasopismach komputerowych.

Co jest powodem niedziałania niektórych programów na ACDTV i czy można temu zaradzić?

Główną przyczyną niedziałania niektórych programów na ACDTV jest zastosowany w niej nietypowy ROM. Jego "inność" związana jest z koniecznością obsługi i sterowania wbudowanego w komputer napędu CD-ROM. Standardowo zmiany te nie powinny wpływać na działanie programów, jednak przy korzystaniu z zewnętrznej stacji dysków (a nie z kompaktu, do pracy z którym komputer ten zasadniczo został stworzony) pojawiają się kłopoty. Ogólnie trudno jest stwierdzić, inaczej niż metodą prób i błędów, które programy i gry nie będą prawidłowo pracować. W rejonie podwyższonego ryzyka znajdują się przede wszystkim gry typu "Not a DOS" oraz użytki korzystające z tzw. kruczków programowych lub sprzętowych.

Jeżeli chodzi o poprawę takiego stanu rzeczy, spotkałem się ostatnio z ogłoszeniami, w których szereg firm proponowało sprzętową przeróbkę powodującą "prawie 100%" zgodność z A500. Trudno w chwili obecnej ocenić na ile informacja ta jest prawdziwa, gdyż nie miałem do tej pory z czymś takim kontaktu. Jeżeli rzecz zostanie w pełni sprawdzona postaram się o tym napisać.

Czy Amiga CDTV posiada wbudowany kontroler SCSI?

Niestety. Standardowa ACDTV nie jest wyposażona w kontroler SCSI. Istnieje jednak możliwość rozbudowy komputera o ten element w postaci specjalnej przystawki zaopatrzonej w port SCSI i pozwalającej na podłączenie dysku twardego, zarówno wewnętrznego jak i zewnętrznego (szyna danych jest taktowana na 16 MHz).

Czy do A1200 da się podłączyć dysk twardy AT-Bus od PC (3,5")?

Da się. Sprawa ta była już dwukrotnie omawiana na łamach "Amigowca" w numerach 7-8/93 i 9-10/93. Proponuję bliżej zapoznać się z tymi artykułami, a wszystko stanie się jasne.

Czy istnieje sprzętowy upgrade zmieniający CDTV ■ CD32?

Nic takiego nie istnieje i nie liczyłbym na zmianę tej sytuacji. Amiga CD32 jest komputerem w pełni 32-bitowym (ACDTV 16-bitowym) posiadającym nowe kości graficzne (AGA) oraz nowy ROM zawierający system 3.0. Przekształcenie ACDTV w Amigę CD32 wymagałoby, praktycznie rzecz biorąc, wymiany całego wnętrza. Byłoby to nie tylko bezsensowne, ale i niezwykle kosztowne.

Jak podłączać dysk 3,5" do gniazda kontrolera 2,5"? Co oznaczją kropki w tabeli dołączonej do artykułu "Twardy dysk do Amigi"(Amigowiec nr7-8/93)?

Twarde dyski 2,5" różnią się od 3,5" zarówno rozmiarem jak i połączeniem z płytą kontrolera. Dysk 3,5" jest połączony dwoma kablami - cztero-żyłowym doprowadzającym zasilanie 5V i 12V oraz czterdziesto-żyłowym złączem z kontrolerem. Ta druga wiązka jest wykorzystywana do wymiany danych pomiędzy kontrolerem ■ koputerem. W przypadku dysków AT-bus większa część kontrolera uieszczona jest w dysku, ■ kontroler na płycie.

Dyski 2,5" wykorzystują pojedynczą taśmę przewodów, która przesyła zarówno dane, jak i napięcie zasilające - tylko +5V. Na szczęście posiada ona identyczne 40 żył jak w kontrolerze 3,5", ■ dodatkowe 4 kabelki są "dolepione" dopiero na samym skraju gniazda. Naszym zadaniem podczas montażu jest połączenie podstawowych żył komunikacyjnych, z ominięciem 2,5" dodatków.

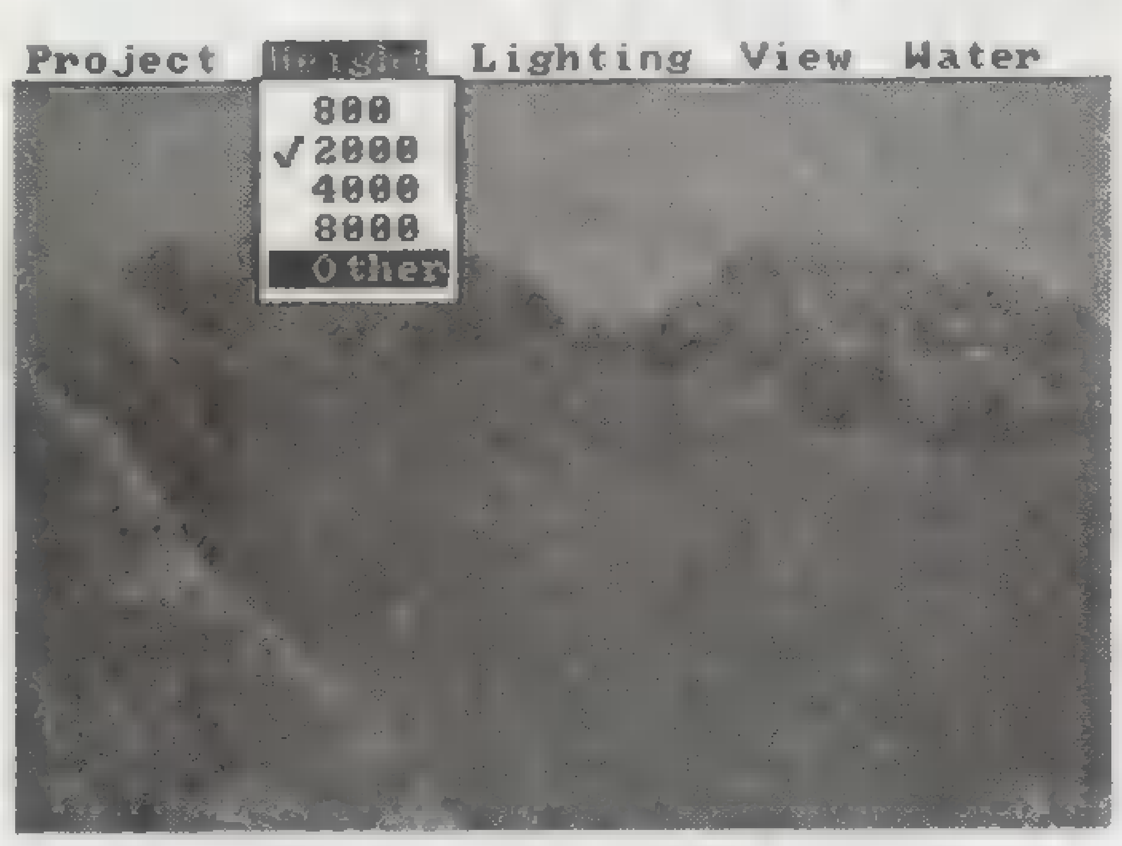
Problem "kropek" w tabelce wynika z przyjętego sposobu liczenia linii, który jest odmienny niż zastosowany w większości gniazd i podstawek. Autorzy zaproponowali liczenie zgodne właśnie z tradycyjnymi metodami, licząc nogi wzdłuż jednej krawędzi, a następnie wzdłuż drugiej (można je również liczyć naprzemiennie, wtedy wystarczy podłączyć pierwsze 40 nóg 2,5" z 40 nogami 3,5", ignorując ostatnie 4 styki). W tym przypadku łączymy pierwsze (licząc od kropki na płycie) 20 nóg złącza 2,5" z jedną stroną złacza 3.5", następnie przeskakujemy cztery styki 21/22/23/24, i kolejne 20 nóg z drugą stroną złącza 3,5". Sposób liczenia nie ma wpływu na połączenie, ważne jest tylko zignorowanie ostatnich 4 styków na płycie (najbliższe złączu monitora).

3,5"- 2,5"
1 - 1
2 - 2
3 - 3
...
19-19
20-20
<- pominięcie 4 styków
21-25
22-26
...
39-43
40-44

Public Domain 2/94

Znowu kolejna dyskietka Public Domain, a w niej jak zwykle dawka programów użytkowych, coś do oglądania, coś do słuchania i coś do zabawy. Tym razem będzie trochę fraktalowo, ale przecież fraktale są teraz w modzie. No, dość ględzenia. Oto co znajdzie się na naszej dyskietce.

1. SCENERY V1.0



Program ten, to zabawa z fraktalami, czyli rysunkami generowanymi przez komputer na podstawie algorytmów matematycznych. Tym razem możemy za pomocą naszej "Amigi" obejrzeć sobie wspaniałe górskie krajobrazy, wzbogacone jeziorami, rzekami i drzewami. Opcje programu wywołujemy z górnego menu.

Oto ich dokładny opis:

About - Informacje o programie i autorze.

Generate Scene - Tutaj zmuszamy komputer do generowania obrazka. Opcja ta posiada dwa warianty, dzięki którym możemy generować krajobrazy z użyciem aktualnego parametru generacji (Use same seed), bądź wybranego przez nas (Use new seed) z zakresu 1-99999.

Redraw Scene - Działa podobnie jak opcja opisana wyżej, z tą różnicą, że zaoszczędzamy 30 sekund na wstępne obliczenia. Komendy tej możemy używać tylko wtedy, gdy nie zmienialiśmy parametru Seed lub Height.

Hide Title - Chowa górną belkę udostępniając naszym oczom cały obrazek.

Save Screen - Zapisuje stworzony krajobraz w formacie Iff 320*200*32.

Stop Drawing - Przerywa generowanie obrazka.

Exit - Wyjście z programu.

Height - Tutaj ustalamy wysokość naszych gór. Można skorzystać ze standardowych wartości, bądź wybierając Other, wpisać daną wysokość.

Lighting - W tym menu wybieramy położenie źródła światła w naszym świecie. Do wyboru mamy:

Direct Overhead - oświetlenie bezpośrednie, umieszczone za nami.

Behind Right Shoulder - światło padające z prawej strony.

Behind Left Shoulder - światło padające z lewej strony.

Front (no shadows) - obraz będzie równomiernie oświetlony i pozbawiony cieni.

Back Light - światło świeci nam po oczach, a góry dają zaciemniające krajobraz cienie.

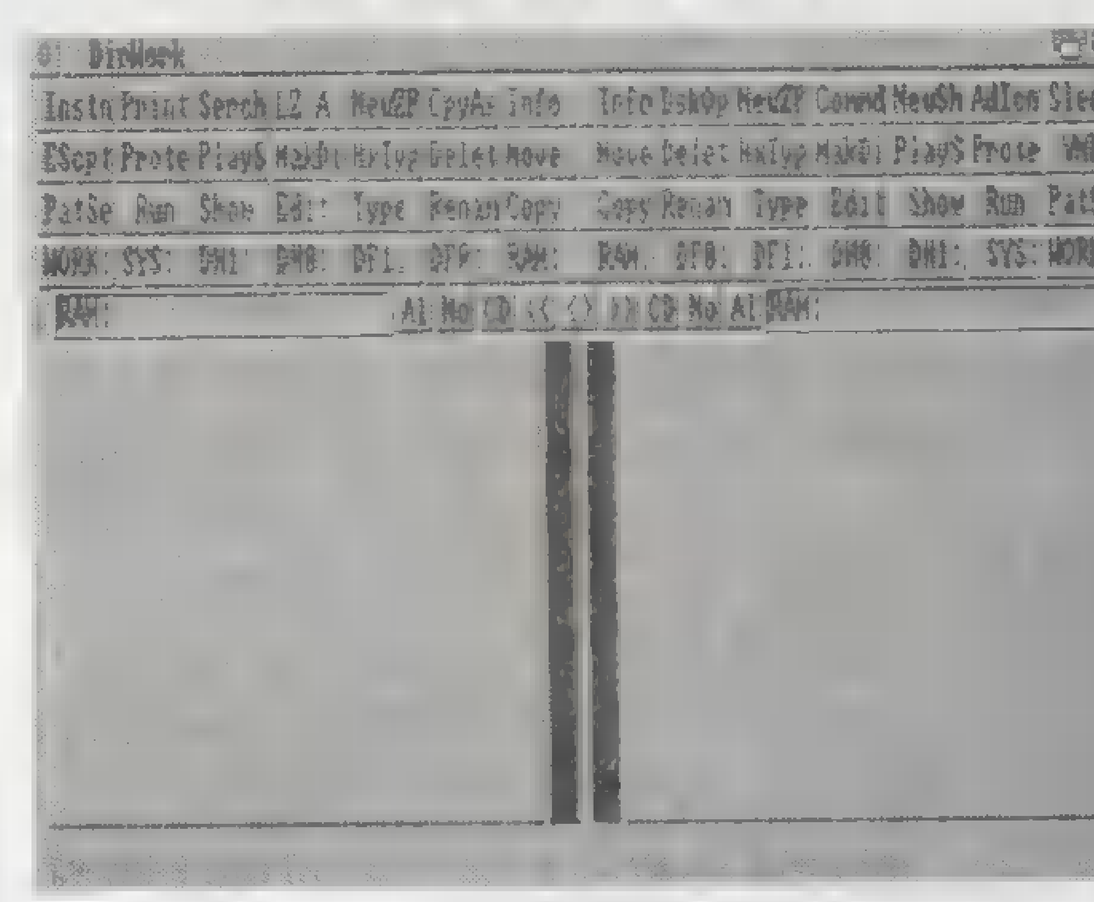
View - Tutaj zmieniamy pole widzenia, a więc wysokość obserwatora.

Water - Tą komendą ustalamy ilość wody w naszym świecie. Możemy wybierać od całkowitego jej braku (None), do istnej powodzi (High).

Generowanie krajobrazu trwa kilka minut, a na ekranie możemy na bieżąco obserwować powstawanie naszego świata. Eksperymentując można uzyskać naprawdę wspaniałe widoki. Może brakuje im trochę do tych z Reala V2.0, ale napewno są satysfakcjonujące.

Autorem programu jest Brett Casebolt.

2. Dir Work V1.51



To kolejny program typu Filemaster, który ułatwia pracę z plikami. Spośród wielu podobnych programów Dir Work wyróżnia się dużą ilością funkcji i sposobem konfiguracji. Program daje się dowolnie konfigurować, dzięki czemu wydaje się być bardzo uniwersalnym narzędziem. Dir Work jest głównym składnikiem całego pakietu, do którego należą jeszcze DWEdit - obszerny program do ustawiania konfiguracji Dir Worka oraz Virus Memory Killer (VMK), którego Dir Work wykorzystuje do testowania pamięci na obecność wirusów. A teraz szczegółowy opis funkcji.

Zacznijmy od skonfigurowania programu. Można to zrobić w dwojaki sposób: pierwszy polega na użyciu zewnętrznego edytora (DWEdit) i daje bardzo duże możliwości, natomiast drugi sposób, to konfiguracja programu za pomocą jego własnego menu. Na wstępie zajmijmy się drugim sposobem. Opcje konfigurujące umieszczone są w górnym menu (DWStuff) uaktywnianym prawym klawiszem myszy.

Buffered Directories - uaktywniając tę opcję włączamy buforowanie katalogów. Daje to duże oszczędności czasu, gdyż przy powracaniu z aktualnego

katalogu do katalogu nadrzędnego, program nie musi wykonywać odczytu jego zawartości (została ona wcześniej zapamiętana).

Fast & Smooth Floppy Dirs - włączenie opcji szybkiego odczytu katalogu. Nie wiem jak autor tego dokonał, ale odczyt katalogu jest około dwukrotnie szybszy!

Auto Floppy Change Get Dir - uaktywnienie tej opcji powoduje, że z każdej włożonej do stacji dyskietki zostanie automatycznie odczytany katalog.

Check Disks For Bootblock Viruses - gdy funkcja ta jest aktywna, po włożeniu dyskietki do stacji, program odczyta jej bootblock i porówna ze standardowym bootblockiem AmigaDos. W razie niezgodności, na ekranie zostanie wyświetlone okno z zawartością bootblocku oraz dwoma gadżetami:

Instal - nagranie standardowego bootblocku AmigaDos.

Cancel - rezygnacja.

Periodic Virus Memory Check - funkcja powoduje włączenie powtarzającego się okresowo testu pamięci (korzysta z VMK). Okres testowania można zmienić programem DWEdit.

Screen Blanker - opcja wygaszania ekranu przy dłuższym spoczynku komputera. Czas po jakim zostanie wygaszony ekran ustala się programem DWEdit. Opcja pomaga ochronić luminofor naszego monitora przed wypaleniem. Gdy zapomnimy wyłączyć sprzęt i pójdziemy oglądać film lub zadzwoni do nas kumpel i zacznie opowiadać nowo zdobyty program, blanker samoczynnie wygasi ekran.

Dirs Above Files - włączenie tej opcji powoduje, że odczytane katalogi będą wyświetlane w okienku zawartości dysku nad plikami.

Display .info Files - wyłączenie tej opcji objawi się tym, że w oknie zawartości dysku nie będą wyświetlane pliki z rozszerzeniem .info, czyli ikony.

Close Gadget=Sleep - opcja ta zmienia funkcję gadżetu wyjścia z programu. Jej uaktywnienie spowoduje, że kliknięcie na wspomniany gadżet nie spowoduje wyjścia z programu, ale go uśpi. Powrotu z uśpienia dokonujemy kombinacją klawiszy Ctrl+Shift+Alt+D.

Screen - dzieki tej opcji możemy dokonać wyboru na jakim ekranie ma uruchomić się program, tzn. na swoim własnym (Custom Screen) czy na ekranie Workbencha (Workbench Screen).

I to by było na tyle co do konfiguracji. Pora teraz na opis innych funkcji menu DWStuff.

Use External Configuration Editor - opcja ta powoduje uruchomienie programu DWEdit, który jest zewnętrznym edytorem konfiguracji.

Change Last Action/Button - tutaj możemy zmienić sposób działania poszczególnych gadżetów programu. Najpierw klikamy na dany gadżet, a następnie wybieramy opisywaną tu opcję. W wyniku otrzymujemy okienko, za pomocą którego możemy zmienić funkcję wybranego gadżetu.

Change Popkeys/Tools Data - jak zauważyliście obok menu DWStuff znajduje się jeszcze menu Tools, w którym możemy znaleźć definiowane przez nas polecenia. Za pomocą tej funkcji możemy edytować polecenia menu Tools lub dodawać nowe. W przypadku Kickstartu powyżej 1.3 polecenia z menu Tools będą dodawane do menu Tools Workbencha.

Save Current Setup To a DW - funkcja zapisu aktualnych ustawień do pliku DW. Program wyświetla okienko ze ścieżką dostępu do DW, co daje możliwość skorygowania miejsca nagrania naszego pliku konfiguracyjnego. Całą operację możemy zatrzymać klikając na Cancel.

Teraz trochę o tym co widzicie. Ekran programu jest podzielony na trzy części. Licząc od góry są to: gadżety, pod nimi dwa okna wyświetlające zawartość dyskietki, a na samym dole znajduje się linia, na której znajdziecie nazwę aktualnej dyskietki, informacje o zabezpieczeniu dysku przed zapisem (WProtected), ilości wolnego miejsca na dyskietce oraz ewentualnie ilości i łącznej długości zaznaczonych plików lub katalogów.

Opis gadżetów.

Instn - zastępuje kliknięcie na pliku.

Print - wysyła zaznaczony plik na drukarkę.

Search - wyszukuje w wybranym pliku podanego ciągu znaków. Kliknięcie na ten gadżet powoduje wyświetlenie okna służącego do wpisania ciągu znaków do wyszukania. W oknie tym znajduje się także gadżet UC==LC, który ustala, że program będzie równoznacznie traktował duże i małe litery. Ponowne kliknięcie powoduje wypisanie "UC!=LC", co oznacza, że podczas wyszukiwania będą rozróżniane małe i duże litery.

LZ A - gadżet ten obsługuje archiwizer LZ.

NewZP - działania tej komendy nie mogłem sprawdzić, gdyż wymaga ona istnienia w katalogu C programu NewZAP, który jest najprawdopodobniej edytorem plików.

Info - powoduje wyświetlenie w dolnej lini ekranu aktualnych informacji o dyskietce.

Commd - komenda ta uaktywnia okno CLI. Powrót następuje przez wpisanie ENDCLI lub naciśnięcie gadżetu Close w lewym górnym rogu.

NewSh - działa podobnie jak komenda opisana wyżej, lecz otwiera okno Shella.

AdIcn - polecenie to powoduje dodanie ikony odpowiedniego typu do zaznaczonego pliku bądź katalogu.

Sleep - Usypia program. Powrót przez kombinację Ctrl+Shift+Alt+D.

EScpt - Działa jak komenda EXECUTE z CLI, to znaczy wykonuje plik skryptowy.

Prote - zakłada zaznaczonym plikom atrybut UNDELETE, w przypadku, gdy atrybut taki jest już założony zostaje on usunięty.

PlayS - polecenie powoduje odegranie zaznaczonego sampla.

Type - powoduje wyświetlenie zaznaczonego pliku tekstowego w postaci znaków ASCII. Jeżeli plik jest skompresowany, zostanie automatycznie rozpakowany. Plik jest wyświetlany w niezależnym oknie. Mamy tutaj do wyboru kilka funkcji dostępnych przez dolne gadżety, bądź z klawiatury:

Next Line, j lub 2 z klawiatury numerycznej - przewinięcie tekstu o jedną linię do góry.

Prev Line, k lub 8 z klawiatury numerycznej - przewinięcie tekstu o jedną linię w dół.

Next Page, Space lub prawy Shift+strzałka w dół lub 3 z klawiatury numerycznej - wyświetlenie następnej strony tekstu.

Prev Page, Backspace lub prawy Shift+strzałka w górę lub 9 z klawiatury numerycznej - wyświetlenie poprzedniej strony tekstu.

Hex/ASCII, h - zmiana sposobu wyświetlania: w znakach ASCII lub heksadecymalnie.

Search, s - wyszukiwanie w tekście podanego ciągu znaków. Obsługa taka jak przy wyszukiwaniu podanego ciągu w pliku (patrz wyżej - przyp. red.).

Exit, q lub Esc - wyjście.

Funkcje dostępne tylko z klawiatury:

Ctrl+strzałka w górę lub 7 z klawiatury numerycznej - powrót do początku tekstu.

Ctrl+strzałka do dołu lub 1 z klawiatury numerycznej - przeskok na koniec tekstu.

HxType - działanie jak wyżej, tylko zaznaczony plik zostaje wyświetlony w postaci heksadecymalnej. Funkcje dodatkowe jak w Type.

VMK - wczytuje i uruchamia program Virus Memory Killer służący do kontroli wektorów systemowych używanych przez wirusy.

PatSe - działanie tej funkcji najlepiej wyjaśni przykład. Załóżmy, że mamy dyskietkę, na której jest wiele plików, między innymi są również obrazki, które chcemy obejrzeć. Załóżmy również, że obrazki te mają nazwy z rozszerzeniem .pic. Ręczne wyszukiwanie takich plików wśród wielu innych mogłoby zająć dużo czasu. Funkcja ta pozwala na zaoszczędzenie czasu. Obsługa jest prosta: po kliknięciu na gadżet PatSe otwiera się okienko z miejscem na podanie wzorca do zaznaczania. Standardowo podane są dwa znaki: ?# zastępujące dowolny ciąg znaków. Aby zaznaczyć nasze obrazki, do istniejącego w okienku ciągu znaków (?#) musimy dopisać rozszerzenie .pic, tak, aby nasz wzorzec miał postać ?#.pic. Następnie klikamy na GoDoIt lub naciskamy Enter. Zgodnie z przewidywaniami w oknie zawartości dysku zostaną zaznaczone wszystkie pliki z rozszerzeniem .pic. Można dokonać operacji odwrotnej i zaznaczyć wszystkie pliki oprócz .pic. W takim przypadku wzorzec powinien wyglądać następująco: ~(?#.pic). Opisana funkcja działa na maszynach wyposażonych w system 2.0 lub wyższy.

Run - powoduje uruchomienie zaznaczonego pliku. Kliknięcie na ten gadżet spowoduje wyświetlenie okna dialogowego, w którym możemy podać ewentualne parametry dla uruchamianego programu. Gdy wybierzemy Run bez zaznaczenia pliku, to do okna możemy wpisać wybraną nazwę lub komendę CLI.

Show - to polecenie powoduje wyświetlenie grafiki zapisanej w formacie IFF. Wyświetlane będą standardowe obrazki IFF, ■ także te w trybie DHIRES oraz ikony. W przypadku, gdy zaznaczymy więcej niż jeden plik, następny będzie wczytywany do pamięci podczas oglądania poprzedniego!

Edit - komenda edycji plików. Do działania potrzebuje umieszczenia w katalogu C dowolnego edytora tekstu i podania jego ścieżki dostępu.

Copy - polecenie kopiowania zaznaczonych plików lub katalogów. Występuje po prawej i lewej stronie ekranu. Kliknięcie na gadżet po lewej stronie spowoduje skopiowanie wybranych plików bądź katalogów z lewego okna do prawego. Kliknięcie na gadżet po prawej stronie - analogicznie odwrotnie.

CpyAs - działa jak Copy, ale dodatkowo otrzymujemy okno, w którym możemy zmienić nazwę pliku.

Move - komenda służy do przenoszenia plików. Po zakończeniu kopiowania plik zostaje usunięty z miejsca, z którego był kopiowany.

Delete - Służy do kasowania zaznaczonych plików lub katalogów. Uwaga! Przy usuwaniu plików program nie oczekuje na potwierdzenie, trzeba więc zachować ostrożność.

MakDir - Zakładanie katalogów. Po wybraniu tej komendy otwiera się okno, w które należy wpisać nazwę nowego katalogu.

Rename - Tutaj zmieniamy nazwę wybranych plików lub katalogów. Kliknięcie na ten gadżet powoduje wyświetlenie okna, w które należy wpisać nową nazwę pliku bądź katalogu.

Al - Zaznaczenie wszystkich plików w oknie.

No - Odznaczanie wszystkich plików w oknie.

CD - Ustawienie katalogu jako aktualny.

>> - Kopiuje zawartość (spis plików i katalogów) z lewego okna do prawego.

<< - Kopiuje zawartość z prawego okna do lewego

<> - Wymienia zawartość prawego i lewego okna.

DskOp - Operacje dyskowe. Dostępne są:

DiskCopy - kopiowanie całych dyskietek. Niezbędne jest co najmniej 1MB pamięci RAM lub minimum dwie stacje dysków. Aby wykonać kopię dyskietki należy kliknąć na odpowiedni gadżet po lewej stronie wybierając napęd źródłowy (DF0÷DF3) oraz ustawić gadżetem po prawej stonie napęd docelowy. W przypadku posiadania jednej stacji dysków oba gadżety ustawiamy na DF0. Następnie klikamy na DiskCopy. Po wczytaniu dyskietki do pamięci możemy nagrać dane na inną dyskietkę klikając na WriteBuf (operację da się powtórzyć wielokrotnie). Zapis jest prowadzany bez weryfikacji, więc aby sprawdzić poprawność zapisanych informacji wyłączamy gadżety oznaczające napędy docelowe (po prawej) i klikamy na DiskCopy.

Format - formatowanie dyskietek. Formatowane będą dyskietki umieszczone w napędach zaznaczonych jako docelowe. Można dokonać formatowania w Fast lub Old File System. Nazwę dyskietki wpisujemy w okno po prawej stronie (standardowo Blank). Formatowanie odbywa się bez weryfikacji, więc jeżeli nie jesteśmy pewni swojej dyskietki należy sprawdzić poprawność zapisu.

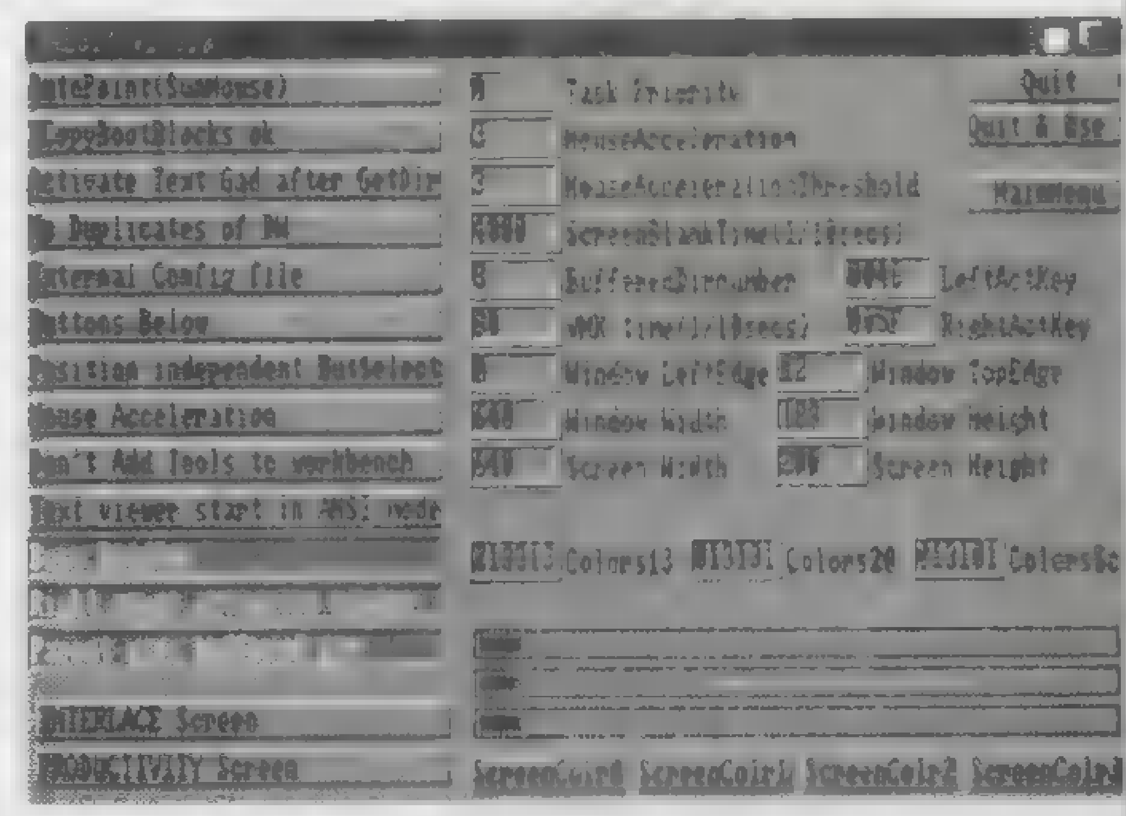
FormatQk - tak zwane formatowanie szybkie, które nie polega na fizycznym skasowaniu danych, lecz na sformatowaniu sektorów, w których znajdują się informacje o danych zapisanych na dyskietce. Funkcji tej nie można stosować do świeżo kupionych dyskietek bądź nie nagranych w systemie AmigaDos.

Install - funkcja zapisuje na dyskietce, ustawionej jako docelowa, standardowy bootblock.

A także: Powrotu z katalogu aktualnego do nadrzędnego dokonujemy klikając na symbol "/" znajdujący się obok okienka ze ścieżką dostępu. Najeżdżając na to okno I przytrzymując prawy guzik myszki, otrzymujemy listę ścieżek dostępu, z której możemy wybrać tę, która nas interesuje. Pojedyncze kliknięcie prawym guzikiem myszy na wspomniane okienko spowoduje odczytanie zawartości aktualnego katalogu.

Autorem programu jest Chris Hames.

3. DWEdit



Program DWEdit jest kolejnym składnikiem pakietu Dir Work. Służy on do ustalania konfiguracji DirWorka i posiada znacznie większe możliwości niż funkcje konfiguracyjne zawarte w samym programie. Niedoświadczonym użytkownikom nie radzę zbytnio kombinować, gdyż tym programem można całkowicie zmienić działanie DirWorka, a nie byłoby wskazane, aby pod komendą Type kryło się kasowanie plików. Opis DWEdit nie będzie zbyt szczegółowy jako, że skierowany jest do bardziej doświadczonych użytkowników, których zachęcam do wgłębienia się w angielskojęzyczny DWEdit.doc.

Gadżet Options - tu ustawiamy parametry związane z komunikacją programu z użytkownikiem:

Autopoint - powoduje automatyczne uaktywnienie okna, pod którym znajduje się kursor myszy.

XCopy Bootblocks Ok - podczas testu bootblocku na obecność wirusów, bootblock zakładany przez Xcopy będzie traktowany jako niegroźny.

External Config File - program będzie korzystał z zewnętrznego pliku zawierającego konfigurację. Jest to o tyle ważne, że normalnie konfiguracja jest zapisywana wprost do pliku DW, a to może sprawić kłopoty w przypadku jego skompresowania.

Buttons Below - zmienia wygląd programu przenosząc gadżety na dół, a okna do góry.

Position Independent - gadżety będą reagowały na prawy klawisz myszy.

Mouse Acceleration - przyspieszenie poruszania kursora myszki. Ustawienia prędkości dokonujemy w oknie obok podając wartość 1÷10. Można także zmienić parametr określający czułość, to znaczy przyspieszenie jakie trzeba nadać myszy, aby włączył się akcelerator. Dokonujemy tego wpisując do okienka Mouse Acceleration Threshold wartość 1÷20.

Don't Add Tools to Workbench - włączenie tego gadżetu spowoduje, że opcje z menu Tools DW nie będą dodawane do menu Workbencha.

Text Viewer Start in ANSI - przeglądarka plików będzie wyświetlała je z uwzględnieniem kodów ANSI.

Busy Pointer - gdy program jest zajęty wykonywaniem jakiegoś zadania, kursor myszy zmieni wygląd ze strzałki na zegarek.

Parent Uses Buffered Dir - komenda Parent będzie używała buforowania spisu katalogów.

Interlace Screen - przełączenie ekranu w tryb interlace.

Productivity Screen - przełączenie ekranu na tryb productivity wykorzystujący monitor VGA.

W menu Options możemy ustalić jeszcze następujące parametry:

Task Priority - priorytet przy pracy w multitaskingu.

Screen Blank Time - czas, po jakim zostanie wygaszony ekran. Podajemy go w 1/10 sekundy.

VMK Time - okres, co jaki pamięć będzie testowana programem VMK. Podajemy w 1/10 sekundy.

Możemy także ustalić wymiary i położenie ekranu oraz jego kolory.

Gadżet Ed PopTools - tutaj można zmienić bądź dodać nowe komendy do menu Tools. Jak już wspominałem komendy te zapisywane są w pewnym kodzie, którego opis zająłby zbyt wiele miejsca, aby go tutaj przedstawić, lecz i na to jest rada. Na ekranie znajdują się gadżety informujące o znaczeniu znaków kodu i ogólnych zasadach jego zapisu, a poza tym mamy przecież "doca".

Oprócz wyżej wymienionych, na ekranie głównym znajdują się jeszcze gadżety z nazwami obsługiwanych rozszerzeń. Kliknięcie na któryś z nich powoduje otwarcie ekranu, na którym możemy dokonać zmiany programu obsługującego pliki z danym rozszerzeniem.

Autorem programu jest Chris Hames.

4. VMK

Virus Memory Killer - to kompletna nazwa tego programu. Jest on również składnikiem pakietu DirWork, ale może być używany osobno. Jego działanie polega na sprawdzaniu wartości wektorów systemowych i informowaniu o ich zmianach.

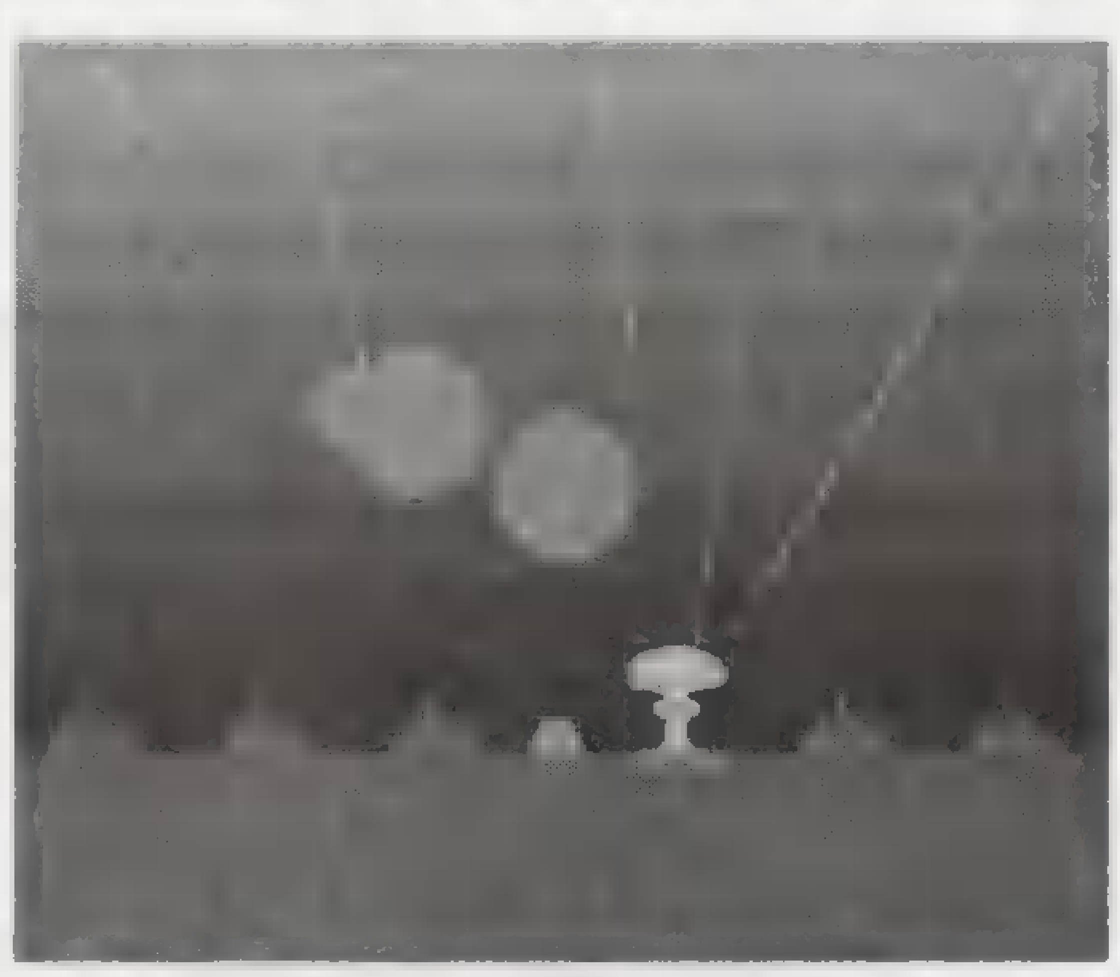
Autorem programu jest Chris Hames.

5. Zoom Windows

Ten programik urozmaici nam pracę z Workbenchem oraz programami wykorzystującymi okienka. Jego działanie objawia się tym, że w momencie otwarcia nowego okna nie pojawia się ono natychmiast, lecz rośnie do normalnych rozmiarów rozpoczynając od aktualnego położenia kursora myszy. Podobnie odbywa się zamykanie okienka.

Autorem programu jest Dr. Maybe

6. Missile Command



To amigowa wersja gry znanej z salonów. Notabene, o ile pamiętam była to jedna z pierwszych gier, których nie obsługiwało się joystickiem, ale za pomocą śmiesznej kuli zamontowanej na pulpicie (track ball). Ale wracając do tematu. Było cicho. Sześć miast i centrum obrony rakietowej spało spokojnie. Nagle zawyły syreny. Atak nuklearny! I tutaj wkraczasz ty. Odważny, dzielny, niezwyciężony. Wcielasz się w walecznego dowódcę obrony rakietowej Ziemi. Na nudę nie narzekasz, gdyż z nieba nieustannie lecą nuklearne podarunki pozostawiając po sobie świetliste smugi. Chwytasz za myszkę, celujesz i wysyłasz w niebo grad antyrakiet. Kuliste obłoki wybuchów pochłaniają wrogie pociski. Co jakiś czas z nieba spadają złośliwe bomby samonaprowadzające, z nimi jest trochę więcej roboty. Musisz ciągle panować nad sytuacją, chwila nieuwagi i Twoje miasto w jednej chwili zamieni się w obłoczek pary, a na jego miejscu wyrośnie niejadalny grzybek. Zwracaj uwagę na ilość amunicji jaka Ci pozostała (kropki na centrum obrony), jej brak to zagłada dla świata.

Gra jest napisana w assemblerze. Posiada przyjemną oprawę graficzną i efekty dźwiękowe. Możliwa jest praca w multitaskingu, ale inne programy nie mogą wydawać z siebie dźwięków, gdyż może skończyć się to upadkiem systemu. Na wszelki wypadek możemy wyłączyć efekty dźwiękowe klawiszem S. Przełączania ekranów dokonujemy Lewa Amiga + N lub M. Esc powoduje zakończenie gry.

Pamiętaj tylko Ty możesz uratować świat ! Liczymy na Ciebie !

Autorem programu jest Max Bithead.

7. Oprócz wyżej wymienionych programów na dyskietce znajduje się moduł muzyczny Light in eden, napisany przez Dagona z Zero Zone oraz grafika niejakiego Bobo z tej samej grupy.

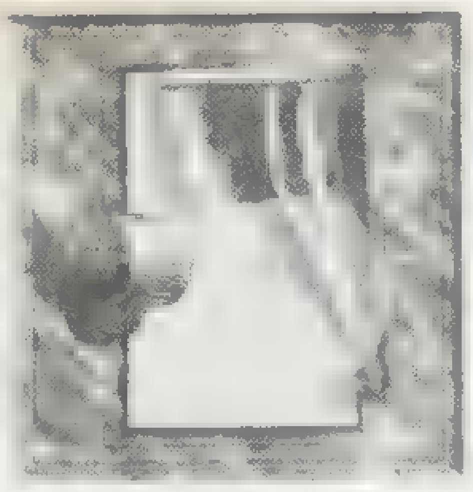
8. Listingi programów. W tej szufladzie znajdziecie listingi do wszystkich kursów języków programowania prowadzonych w Amigowcu

Autorem większości ikonek zamieszczanych na naszych dyskietkach jest Dariusz Zwierzyński.

Prosimy o uwagi o naszej bibliotece. Piszcie do nas, jakich programów poszukujecie i jakie powinny się w niej znaleźć. Czekamy też na owoce Waszej pracy.

Jak zamawiać i ile to kosztuje ?

Dyskietki PD można zamawiać przesyłając pieniądze na nasze konto, przy pomocy blankietu z Amigowca, podając symbol dysku PD (od #1/92 do 12/92, #1/93, #2/93, #3-4/93, #5-6/93, #7-8/93 oraz "A", "B" i "C") i oczywiście swój czytelny adres. Nasza dyskietka kosztuje 35 tysięcy złotych (wliczone są koszty dyskietki, przesyłki, nalepki, opakowania itp.). W przypadku przesyłki za zaliczeniem pocztowym, zamawiający pokrywa koszty zaliczenia. ❑



LISTA DEMONÓW

Artur Łukasik, Sebastian Kłomski

Witamy po raz szósty! Tym razem zmiany na naszej liście. Dotyczą one właściwie tylko listy zagranicznej, choć i na polskiej scenie coś drgnęło. Jest to wynikiem wypuszczenia przez wiele grup (w tym kilku polskich) swoich produkcji na Party III. Było to największe party w historii światowej sceny. Odbyło się w dniach 27-29.12.1993. w Herning w Danii.

ZAGRANICA

GRUPY:

1. ***Sanity***
2. ***Kefrens***
3. ***Virtual Dreams*/Fairlight**
4. ***Complex***
5. ***Spaceballs***
6. ***Lemon***
7. ***Melon Dezign***
8. ***Movement***
9. ***Silents***
10. ***Andromeda***

Po Party III w Danii nastąpiło dość znaczne przetasowanie. Grupa Sanity udowodniła, że należy się jej miano najlepszej na świecie. Na miejscu drugim chyba niezagrożona grupa Kefrens, która już od dłuższego czasu pracuje nad znakomitym demem na party w Danii (Gathering). Oczywiście wersja na Amigę 1200. Na dalszych pozycjach grupy Virtual Dreams/Fairlight oraz Complex, które na wyżej wspomnianym party zajęły pierwsze (Complex) i drugie (Fairlight) miejsca.

DEMA:

1. ***Hardwired*/Cronics & The Silents**
2. ***Desert Dream*/Kefrens**
3. ***Arte*/Sanity**
4. ***9 Fingers*/Spaceballs**
5. ***Extension*/Pygmy Projects**
6. ***Origin*/Complex**
7. ***Full Moon*/Fairlight**
8. ***World of Commodore*/Sanity**
9. ***State of The Art*/Spaceballs**
10. ***Guardian Dragon II*/Kefrens**

Ostatnio pojawiło się wiele nowych dem. Warto zwrócić uwagę na fakt, że Origin/Complex i Full Moon/Fairlight pomimo tego, iż zajęły dwa pierwsze miejsca na Party III w Danii, zajmują niższe lokaty na naszej liście niż dema Arte/Sanity i 9 Fingers/Spaceballs. Spowodowane jest to tym, że w Polsce bardziej popularna jest A500 niż A1200.

DEMA PLIKOWE:

1. ***Interference*/Sanity**
2. ***Mindriot*/Andromeda**
3. ***Paradigma*/Complex**
4. ***3D Demo II*/Anarchy**
5. ***Announce*/Lemon**
6. ***Clairvoyance*/Absolute**
7. ***Transpiersky*/The Silents**
8. ***Terminal Fuckup*/Sanity**
9. ***Optimum Fuckup*/Sanity**
10. ***Joyride*/Phenomena**

INTRA 40 KB:

1. ***Tetris*/Melon Dezign**
2. ***Virtual Dreams***
2. ***Shining***
3. ***Lemon***
4. ***Fairlight***
5. ***Arrilfoll*/Dual Crew & Shinig**
6. ***Stellar***
7. ***No Pain No Gain*/TRSI**
8. ***The Silents***
9. ***Essence***
10. ***Spaceballs***

MUSIC DISKI:

1. ***Memorial Songs*/Alcatraz**
2. ***Jesterday*/Sanity**
3. ***Chromagic*/ Dual Crew & Shining**
4. **Mirror/Andromeda**
5. ***Crystal Symphonies II*/ Phenomena**
6. ***Turmoil*/Sanity**
7. ***Kuglep0len*/Static Bytes**
8. ***Book of Song*/Complex**
9. ***Music Dream II*/Phenomena**
10. ***Crystal Symphonies I*/ Phenomena**

KODERZY:

1. ***Chaos*/Sanity**
2. ***Laxity*/Kefrens**
3. ***The Spy/Crionics***
4. ***Mr.Pet*/Sanity**
5. ***Tsunami*/Faitlight**
6. ***Lone Starr*/Spaceballs**
7. ***Hannibal*/Lemon**
8. ***Microforce*/Sanity**
9. ***Performer*/Melon Dezign**
10. ***Tron*/Sanity**

GRAFICY:

1. ***Cougar*/Sanity**
2. ***R.W.O.*/Kefrens**
3. ***Facet*/Lemon**
4. ***Ra*/Sanity**
5. ***Mack*/Melon Dezign**
6. ***Peachy*/Masque**
7. ***Rack*/Absolute**
8. ***Havok*/Sanity**
9. ***Walt*/Melon Dezign**
10. ***Titan*/Complex**

MUZYCY:

1. *Jester*/Sanity
2. *Moby*/Sanity
3. *Lizard King*/Alcatraz
4. *Audiomonster*/Melon Dezign
5. *Laxity*/Kefrens
6. *Nuke*/Lemon
7. *Tip & Mantronix*/Phenomena
8. *Jesper Kyd*/The Silents
9. *Virgill*/Masque
10. *Clawz*/Complex

POLSKA
GRUPY:

1. *Old Bulls*
2. *Deform*
3. *Applause*
4. *Alchemy*
5. *W.F.M.H.*
6. *Union*
7. *Suspect*
8. *Joker*
9. *Damage*
10. *TRSI PL*

Na dziesiątej pozycji pojawiła się nowa grupa TRSI PL (jest to polska sekcja znanej na całym świecie formacji Tristar & Red Sector Incorporation). Poniżej podajemy jej skład:

Walt (ex. Investation)
Dr.D (ex. Investation)
Tap (ex. Investation)
Reset (ex. Investation)
Jeżo (ex. Suspect)
Norby (ex. Investation)
XTD (ex. Mystic & Union)
Trash Head (ex. Suspect)

Jak widać grupę tę zasilają głównie ludzie z byłego Investation. Pozostała część Investation pod wodzą The Tradera założyła team pod nazwą Fear. Krążą plotki, że jeden z członków TRSI PL (XTD) zaproponował ludziom z TILTu przystąpienie do jego grupy. TILT stanowczo odmówił motywując swoją decyzję brakiem perspektyw. Warto dodać, że nie tak dawno (a dokładnie parę dni po Polish Autumn'93 w Poznaniu) z podobną propozycją wystąpiło dogorywające Mad Elks. Oczywiście i w tym przypadku odpowiedź była odmowna.

TRSI PL planuje w najbliższym czasie wydać następujące produkty:

dysk muzyczny z muzyką XTD
krótkie demko
slide show.

DEMA:

1. *Impulse II*/TILT/Old Bulls
2. *Deformations*/Deform
3. *Fugazi*/Old Bulls
4. *PRO.s.i.a.k*/FCI
5. *Hallucinations & Dreams*/ Union
6. *Software*/W.F.M.H.
7. *Marchewki*/Alchemy
8. *Technological Death*/ Mad Elks
9. *Uquala*/Damage
10. *The Return*/Joker

Na pierwszym miejscu demo Impulse II/TILT/Old Bulls, które miało zostać wystawione na Party III w Herning w Danii. Demo jednak zginęło w niewyjaśnonych okolicznościach. Nie wiadomo do dzisiaj czy była to wina tylko i wyłącznie organizatorów, czy też człowieka, któremu owe demo zostało powierzone przez autorów. Z kilku niezależnych źródeł wiemy, że Impulse II zajęłoby co najmniej piąte miejsce.

DEMA PLIKOWE:

1. *Partyzanci*/Joker
2. *Bara Bara*/Applause
3. *Impulse-preview*/Old Bulls
4. *X-Mas*/Damage
5. *No Name*/Suspect
6. *Varathron*/Suspect
7. *Autobiography*/Old Bulls
8. *Anam*/Investation
9. *Xenium*/Old Bulls
10. *Defene coś tam*/Luzers

DYSKI MUZYCZNE:

1. *Vengeance*/Beta Team
2. *Orgasm*/Old Bulls
3. *Waiting for Another Beer*/ Joker
4. *Kadi Sound Disk III*/Joker
5. *Hunt for the red kret*/FCI
6. *Illusion*/Old Bulls
7. *Music Tracks*/Suspect
8. *Kadi Sound Disk II*/Joker
9. *Lost Tracks*/Illusion
10. *Sigh*/Old Bulls

KODERZY:

1. *Musashi*/Union
2. *Hudi*/TILT/Old Bulls
3. *Bukka*/TILT/Old Bulls
4. *Robin*/W.F.M.H.
5. *Vico*/Alchemy
6. *Tom*/Union
7. *GBH*/Joker
8. *Falcon*/Applause
9. *Miklesz*/Damage
10. *Kane*/Suspect

MUZYCY:

1. *XTD*/Union
2. *Peters*/Joker
3. *Pic Saint Loup*
4. *Mr.Root*/Union
5. *Accord*/Gel Dezign/Old Bulls
6. *Dreamer*/FCI
7. *Phobos*/Beta Team
8. *Kadi*/Joker
9. *Hudinsky*/Veto
10. *Passat*/Funzine

GRAFICY:

1. *TPP*/Applause
2. *Animal*/Union
3. *Sabe*/TILT/Old Bulls
4. *Seq*/Union
5. *Pluton*/Union
6. *Zefir*/Gel Dezign/Old Bulls
7. *Python*/Union
8. *Frodo*/Alchemy
9. *Rys*/Joker
10. *Berserker*/Investation

MAGAZYNY DYSKOWE:

1. *Zig Zag*/Union
2. *Thing*/Applause
3. *Zasmashka*/Alchemy
4. *Paper White*/Luzers
5. *Fat Agnus*/Investation
6. *Poczytaj Mi Mamo*
7. *Imazine*/Saint Group
8. *X-mag*/Union
9. *Krawężnik*/Status OK
10. *Nie z tej beczki*/Lamers

Nowy wspaniały magazyn grupy Applause o nazwie Thing niespodziewanie pnie się na coraz wyższe lokaty. Dzieje się tak zapewne dlatego, że jest to obecnie najczęściej ukazujący się magazyn dyskowy. Do jego niewątpliwych zalet należy zaliczyć wspaniały design i grafikę oraz bardzo wygodną obsługę. Nie wiadomo czy Applause w najbliższym czasie nie wyprzedzi legendarnego Zig Zaga. Uzależnione jest to tylko w dużej mierze od tego, czy ukaże się wreszcie nowy numer Zig Zaga.

Wszystkie wiadomości pochodzą z pewnych źródeł i są w 100 procentach sprawdzone. Głosujemy na dowolną ilość pozycji z każdego działu, przyznając im od 1 do 10 punktów. Głosy, propozycje, swoje prace, uwagi, a także nowe programy demonstracyjne i magazyny dyskowe prosimy przesyłać na adres redakcji. ❑

KWAŚNO o zasadach.

Jarosław Chrostowski

Poprawne programowanie w jakimkolwiek języku wymaga nie tylko znajomości syntaktyki czy zestawu słów kluczowych. Bez porównania ważniejszą cechą jest umiejętność algorytmicznego patrzenia na problem, który mamy rozwiązać. Umiejętność tę można w sobie wyćwiczyć, jednakże jest ona w znacznym stopniu uwarunkowana wrodzonymi predyspozycjami.

Według hierarchii ważności jako kolejną wymieniłbym umiejętność pisania czytelnego i czystego (wolnego od błędów) kodu źródłowego. Dopiero gdzieś na szarym końcu błąka się znajomość konkretnego języka.

Czytelnicy znają mnie (mam nadzieję...) z dość "technicznych" artykułów wiążących się w mniejszym lub większym stopniu z programowaniem w języku C. Chciałbym tym razem napisać o sprawach nieco bardziej ogólnych, które jednak są nieodłączną cechą dobrego programowania. Nie zamierzam pisać o technice programowania, lecz o rzeczy równie ważnej - właściwej organizacji "pisania" programów. Przy okazji chciałbym zwrócić uwagę czytelników - w szczególności autorów lub potencjalnych autorów artykułów - na pewne sprawy związane ze sposobem przekazywania informacji nie między poszczególnymi taskami, a pomiędzy poszczególnymi osobami. Wydaje mi się, że warto podzielić się czasem takimi uwagami zebranymi podczas kilkunastoletniej już (jak ten czas leci!) pracy z komputerami. Być może zaoszczędzi to komuś trochę czasu. Niektóre z moich uwag związane są z konkretnymi językami programowania, trudno mówić bowiem o nazewnictwie zmiennych automatycznych w basicu, skoro ich tam nie ma. No cóż, ad rem.

Zasada 1: PROGRAM POWSTAJE W GŁOWIE, NIE W KOMPUTERZE.

Siadając do komputera musisz wiedzieć, co chcesz pisać. Niby to takie proste: usiąść i zacząć pisać program. Panatadeuszowskie "Jakoś to będzie...". W efekcie mając program napisany w 80% dochodzimy do wniosku, że cały problem można było rozwiązać inaczej, z reguły cztery razy szybciej i dwa razy bardziej zwięźle. Najlepiej dokładnie przemyśleć przynajmniej zarys programu, jego ogólną budowę, opracować podstawowe problemy techniczne. Jeśli nie masz jasnej wizji tego co chcesz robić, nie rozpoczynaj pisania - stworzysz bajzel a nie program. Lepiej pojechać na ryby (polecam łowienie pstrągów na muchę) lub w miłej, wieczornej atmosferze przy kominku, tuląc się do ukochanej, przemyśleć... ostatnie szczegóły formatu zapisu danych. Jeśli nie lubisz rysować schematów blokowych, nawet najbardziej ogólnych, daruj sobie wieczór przy kominku i przeczytaj zasadę 2.

Zasada 2: NAPISANIE KOMENTARZA KOŃCZY PISANIE PROGRAMU.

Już widzę te uśmiechy: zasadę sobie wymyślił, po co w ogóle komentarz? Jak ktoś kiedyś mądrze powiedział, komentarze w programach ma się za darmo. To tyle na temat sensu pisania komentarzy. Wracając do mojej zasady - wyróżnia się ona jednak spomiędzy innych przykazań dotyczących pisania komentarzy. Dużo osób dopisuje komentarze do gotowego programu. Postępuj inaczej. Otóż siadając do pisania jakiejś funkcji NAJPIERW napisz komentarz: co funkcja ma robić, jakiego typu są poszczególne jej parametry i jakie jest ich znaczenie, co funkcja zwraca lub jakie zmienne modyfikuje. W tym momencie dla mnie kończy się programowanie. Teraz wystarczy jedynie przetłumaczyć komentarz na dany język. A mój komentarz prywatny do tej zasady jest taki: łatwiej się pisze, gdy się wie, co ma się pisać. W dawnych czasach programowanie rozpoczynano od żmudnego konstruowania algorytmów. Wierzcie mi, to nie była głupia zasada i warto do niej, choćby w tak zmodyfikowanej formie, wrócić.

Zasada 3: PISZ KOMENTARZE Z KRÓTKIMI OBJAŚNIENIAMI PRZED KAŻDYM WAŻNIEJSZYM BLOKIEM W PROGRAMIE.

Nie musisz jednak robić tego w każdej linijce. Komentarz dobrze jest oddzielić od poprzedniej i następnej linii programu jedną linią wolną. Jeszcze lepiej jest napisać ten komentarz przed napisaniem bloku. Później z reguły nie chce się tego robić.

Zasada 4: PRZY KAŻDEJ SKŁADOWEJ STRUKTURY NAPISZ KOMENTARZ WYJAŚNIAJĄCY, DO CZEGO MA ONA SŁUŻYĆ.

No cóż, docenisz tę zasadę, gdy kod twojego programu przekroczy 300 KB.

Zasada 5: STOSUJ WYRAŹNE WCIĘCIA, TO UŁATWIA ORIENTACJĘ.

Opracowanie jednorodnego systemu wcięć jest rzeczą bardzo ważną i indywidualną jednocześnie. Osobiście polecam następującą zasadę: dwie spacje wcięcia w przypadku pojedynczego rozkazu, np.

```
if ( size < 300 )
  size += 100;
```

i jedną spację w przypadku nawiasu klamrowego rozpoczynającego blok (nawias ten jest oczywiście w nowej linii) oraz dopiero linię niżej przesunięty o kolejną spację pierwszy rozkaz bloku, np.

```
if ( size < 300 )
{
  size += 100;
  loop -= 2;
}
```

UNIKAJ jak ognia zapisu, w którym dopełniające nawiasy klamrowe nie występują w tych samych kolumnach i nie są same w linii, np.

```
if ( size < 300 ) {
  size += 100;
  loop -= 2;
}
```

Przed każdą instrukcją if, for, while, do...while warto zostawić linię wolną, np.

```
loop += 5;

while ( loop < 100 )
{
  ...
}
```

Skoro mowa o pustych znakach, polecam stawianie spacji przed nawiasami i operatorami. Zresztą porównajmy dwie linie niżej:

```
(((++x+10)/2<20)||( x*(y+z++)))
( ( ( ++x + 10 ) / 2 < 20 ) || ( x * ( y + z++ ) ) )
```

Zasada 6: NADAWAJ ZMIENNYM ZNACZĄCE NAZWY I UŻYWAJ WEDŁUG ZNACZENIA. NAZWY FUNKCJI RÓWNIEŻ POWINNY BYĆ ZNACZĄCE.

Lepiej jest zmiennej nadać znaczącą nazwę, np. zmiennej indeksowej można nadać nazwę licznik lub counter. Osobiście denerwują mnie polskie słowa (np. nazwy zmiennych) przemieszane z angielskim tekstem (słowa kluczowe), stąd zmiennym i funkcjom nadaję nazwy angielskie - przynajmniej mam motywację do przewertowania słownika. Staraj się wykorzystywać zmienne zgodnie z ich nazwą, unikaj wielokrotnego wykorzystania jednej zmiennej do kilku zastosowań. No, chyba że jest to wymagane ze względu na zwartość kodu wynikowego.

Zasada 7: DEKLARACJE ZMIENNYCH AUTOMATYCZNYCH W FUNKCJACH POGRUPUJ W TEN SPOSÓB, BY ZMIENNE TEGO SAMEGO TYPU BYŁY OBOK SIEBIE.

Deklaruj zmienne automatyczne tego samego typu zawsze obok siebie, przy czym deklaracje dobrze uporządkować według "rosnącego" rozmiaru, tj. BYTE, UBYTE, WORD, UWORD, LONG, ULONG, wskaźniki, tablice itp. Jeśli konieczna jest deklaracja nowej zmiennej takiego typu, jak któraś z wcześniej zadeklarowanych danych, wtedy dopisz ją do deklaracji wcześniejszych zmiennych lub, jeśli ci na tym zależy (chcesz opisać deklarację komentarzem itp.), wstaw ją linię niżej, np.

```
UBYTE brud, test, indeks;
LONG suma, różnica;
LONG całka;            /* To jest całka, jakby ktoś nie wiedział. */
```

Zasada 8: NAZWY ZMIENNYCH ZEWNĘTRZNYCH PISZ ROZPOCZYNAJĄC KAŻDY CZŁON Z DUŻEJ LITERY. NAZWY ZMIENNYCH AUTOMATYCZNYCH PISZ MAŁYMI LITERAMI.

To prosta, lecz bardzo ułatwiająca życie zasada. Zmienne zewnętrzne nazywaj np. LicznikBłędów, automatyczne: licznikbłędów. Łatwiej będzie wtedy ustalić na pierwszy rzut oka zakres występowania takiej zmiennej.

Zasada 9: POJEDYNCZE ROBOCZE ZMIENNE ZEWNĘTRZNE GRUPUJ W JEDNĄ STRUKTURĘ.

Często mamy do czynienia z dużą ilością zmiennych zewnętrznych, tj. zmiennych widzianych przez wszystkie funkcje programu. Trudno niekiedy ustalić, szczególnie przy większych programach, co wpływa na jaką zmienną. Zwykle zmienne takie mają także nieprzyjemną tendencję pojawiania się (w sensie deklaracji) w różnych miejscach programu, stąd trudno na pierwszy rzut oka znaleźć je wszystkie. Dobrze jest więc zgrupować je w jedną strukturę, najlepiej o stałej w każdym programie nazwie, np. Status. Zawsze będziesz wiedział, gdzie czego szukać. Znikną kłopoty ze zgraniem aktualnego stanu programu. Polecam stosować w połączeniu z zasadą 4.

Zasada 10: DO TYCH SAMYCH RZECZY UŻYWAJ W PROCEDURACH ZMIENNYCH O TYCH SAMYCH NAZWACH.

Konsekwentne stosowanie tych samych nazw w różnych funkcjach zwiększa czytelność programu I ułatwia orientację, np. zmienna indeksująca pętle w każdej funkcji może się nazywać loop, itp.

Zasada 11: NIE PISZ ROZBUDOWANYCH FUNKCJI, STARAJ SIĘ PODZIELIĆ JE NA SZEREG WYWOŁAŃ FUNKCJI POMOCNICZYCH.

Jeśli funkcja jest zbyt długa, np. kod przekracza 100 linii, wtedy poszczególne bloki takiej funkcji warto zapisać jako funkcje. W wielu wypadkach, szczególnie w początkowej fazie pisania programu, ten prosty zabieg znacznie przyspieszy napisanie funkcji ze względu na fakt, że praktycznie całe ciało funkcji jest widoczne na jednym ekranie. Jeśli uważasz, że w danej funkcji poszczególne bloki programu są na tyle proste i czytelne, że gdy po 10 latach sięgniesz po kod i będziesz wiedział, o co chodzi, wtedy możesz zrezygnować z tej zasady. W przypadku, gdy szacujesz okres identyfikacji na 9 lat, lepiej podziel funkcję.

Zasada 12: FUNKCJOM POMOCNICZYM NADAWAJ NAZWY ZAWSZE WEDŁUG TEGO SAMEGO SCHEMATU.

Tak jak ze zmiennymi, tak również w tym przypadku od razu będziesz wiedział, z czym masz do czynienia np. HelpForSearch(), HelpForReplace(), itp. Zastanawianie się co poeta chciał powiedzieć zostawmy na lekcje historii literatury, szumnie nazywane w liceach językiem polskim.

Zasada 13: ROZBUDOWANE PROGRAMY DZIEL NA MODUŁY, KTÓRE ZAWIERAJĄ FUNKCJE DOTYCZĄCE ZBLIŻONYCH ZAGADNIEŃ.

Również w tym przypadku łatwiej będzie Ci się pisało i sprawdzało program, gdy wszystkie dane i funkcje dotyczące danego tematu będziesz miał w jednym pliku. Nie dość, że łatwiej się taki kod przegląda, to podczas kompilacji będziesz czekał na przetworzenie tylko jednego modułu.

Zasada 14: STARAJ SIĘ PISAĆ PROGRAM JAK NAJBARDZIEJ CZYTELNIE, KORZYSTAJ Z NAJBARDZIEJ ZWIĘZŁYCH FORM ZAPISU TYLKO W PRZYPADKU, GDY JEST TO KONIECZNE, NP. ZE WZGLĘDU NA SZYBKOŚĆ.

Dobrze jest, gdy język pozwala na pisanie bardzo zwięzłych wyrażeń, jednak zbytnie ich nagromadzenie utrudnia wyszukiwanie błędów, a nawet, w przypadku zbyt "maniakalnej" ciągoty do skracania kodu powoduje, że pojawiają się nowe błędy.

Zasada 15: POSTARAJ SIĘ NAUCZYĆ CZYTAĆ "PO LITERZE", NIE "PO WYRAZIE".

Bardzo niebezpieczną rzeczą jest nasza umiejętność czytania tekstu nie litera po literze, lecz po "kształcie" wyrazu. Rozpoznawanie przez nasz mózg określonego zawijasa na zasadzie porównania z pamiętanym wzorcem powoduje, że ewentualne literówki stają się bardzo trudne do wykrycia. Dotyczy to nie tyl

ko samych nazw zmiennych czy słów kluczowych (co wykryje kompilator), lecz także oczywistych błędów typu np. '* /' zamiast '*/', czy '+ +' zamiast '++', itp. Takie wyrażenie często jest kompilowane bezbłędnie. Swego czasu zamiast cyfry 0 napisałem literę O, a że rzecz działa się dawno temu na komodorku C64 w basicu, stąd miałem trochę zabawy. Konkretnie dwa dni. Z biegiem lat czas wyszukiwania takich pomyłek wcale się nie skrócił...

Zasada 16: KTOŚ KIEDYŚ WYMYŚLIŁ INSTRUKCJĘ GOTO. ZAPOMNIJ O TYM.

Prawie zawsze da się jej uniknąć (no chyba, że piszesz w basicu). Wyjątek może stanowić wychodzenie z zagnieżdżonych pętli lub pisanie przerwań w C. WYJĄTEK!!!

Zasada 17: JEŚLI CI "NIE WYCHODZI", ODŁÓŻ PRACĘ DO JUTRA. RANO SKOŃCZYSZ W GODZINĘ.

Nie wal w klawiaturę (no, tego nie zawsze da się uniknąć) i nie wyrzucaj komputera na śmietnik, gdy pięćdziesiąty raz próbujesz napisać funkcję i po uruchomieniu widzisz GURU. Daj sobie spokój. Wstań następnego dnia wcześnie rano, a najprawdopodobniej zrobisz całą rzecz w godzinę. To zadziwiająca właściwość naszego mózgu - nie myślisz o problemie, a "sam" ci się przez noc układa w głowie...

Zasada 18: KORZYSTAJ Z GOTOWYCH ALGORYTMÓW.

Nie w każdym przypadku uda nam się wymyślić właściwy dla danego problemu algorytm. W praktyce wiele problemów zostało już zalgorytmizowanych wcześniej, niektóre zresztą są udoskonalane latami przez całe zespoły ludzi. Nie wpadaj w kompleksy, że nie wymyśliłeś "prostego" algorytmu na pół strony. Te pół strony to często wynik wieloletniej pracy. Nie musisz rozumieć każdego użytego algorytmu (już widzę te gromy na moją głowę ze strony nauczycieli matematyki czy fizyki), do wyprowadzenia części z nich używa się dość (lub bardzo) zaawansowanych metod matematycznych. Jeśli odczuwasz głód wiedzy warto poznać zasady, na których opiera się algorytm. Jeśli nie - cóż, chcesz być programistą, nie matematykiem, fizykiem, księgowym, itp.

Zasada 19: PRZYSTĘPUJĄC DO PISANIA DUŻEGO PROGRAMU ZREZYGNUJ Z JĘZYKÓW NISKIEGO POZIOMU.

Nie da się - niestety - pisać czytelnie i szybko w asemblerze. Posłużę się tutaj pewnym - przewrotnym, przyznaję, lecz nie pozbawionym racji - rozumowaniem. Napisanie "programu życia" to zwykle zabawa na długie miesiące, niekiedy lata. Chcąc pisać program w asemblerze, świadomie wydłużamy czas pisania, jak również ilość możliwych do wystąpienia błędów czy stopień skomplikowania kodu. Program napisany np. w C będzie działał dwa razy wolniej, lecz zostanie napisany przez - powiedzmy - dwa lata. W asemblerze warto pomnożyć czas pisania razy dwa. Postęp techniczny, jaki dokona się w tym czasie w sprzęcie, spowoduje, że program pisany w C będziej działał na nowym sprzęcie z taką prędkością, jaką ileś lat temu przewidywaliśmy dla asemblera. Zresztą newralgiczne pod względem szybkości fragmenty kodu można napisać w asemblerze i bez specjalnych problemów podłączyć do kodu pisanego np. w C.

Zasada 20: OTWIERAJĄC NAWIAS NAPISZ OD RAZU NAWIAS DOPEŁNIAJĄCY.

Proste i skuteczne.

Zasada 21: PAMIĘTAJ, PROGRAMOWANIE TO TAKIE ZAJĘCIE, W KTÓRYM NA IDIOTĘ WYCHODZISZ ZAWSZE TY, NIGDY KOMPUTER.

No comments.

Tyle umoralniania. Chciałbym poświęcić trochę uwagi umiejętności posługiwania się językiem polskim podczas przekazywania informacji o komputerach. Ten temat chciałem poruszyć już znacznie wcześniej, bowiem leżał mi na sercu od dawna. Myślę, że mimo paru docinek moi koledzy po piórze wybaczą mi. Poniższy tekst jest pisany bez jakichkolwiek do kogokolwiek aluzji, aczkolwiek na podstawie wychwyconych przeze mnie "białych kruków".

Otóż zastanawiam się - i myślę, że czytelnicy wraz ze mną - jakie zaufanie można mieć do osoby, która pisząc o tak ścisłym przedmiocie wiedzy jak programowanie robi dziesiątki literówek? Korekta co prawda powinna to wychwycić, ale... nie powinna, bo autor, programista, człowiek, który przekazuje innym swą wiedzę nie ma prawa popełniać tego typu błędów. Świadczą one bowiem o jego niechlujstwie! Jak może (tu się przyznaję, denerwuję) autor rozpoczynać kolejną część cyklu o programowaniu od zdania: "Właściwie w poprzednim numerze powiedzieliśmy sobie wszystko." ??? To po co w takim razie artykuł? No ręce opadają. Rozumiem, że tego typu zdanie może napisać laik, ale Wy jesteście podobno specjalistami. Ludźmi, którzy operują wymagającym myślenia i logiki urządzeniem, ba, przekazują swoją wiedzę innym. O ile fakt posiadania przez Was pewnej wiedzy do niczego nie zobowiązuje, to chęć jej przekazywania - tak! I co, w programie byka znajdziecie od razu, ale w "języku programowania" o nazwie polski myślenie nie obowiązuje? Na litość boską, ludzie czytajcie co piszecie! Tak jak program, kilka razy, od dechy do dechy. Dodam od razu, że co niektórym fatalnie mylą się języki programowania. W C nie napiszecie przecież zamiast nawiasu klamrowego begin, ale w polskim już zaalokujecie sobie pamięć. Ten fenomen daje się wytłumaczyć jedynie chęcią zaimponowania laikom swoją wiedzą. Tymczasem w Polsce wystarczy tę pamięć przydzielić, co wypada prawie tak samo jak zaalokować (dobrze, że nie przez dwa 'l'). Zresztą "przydziel pamięć" jakby zdrowiej brzmi od "zaalokuj pamięć". Naprawdę nie musicie na każdym kroku chwalić się swoją znajomością slangu. W pewnym dwustronicowym artykule znalazłem niedawno kilkanaście razy użyte słowo screen, ani razu - ekran. Cholera, niedługo zapomnę pisać po polsku. Amożebytakpisaćbezodstępów,kodwynikowymisięskróci? Aby nie było nieporozumień: nie chcę, by zamiast task pisać zadanie lub program, czy coś a la "zwis męski". Jednak tam, gdzie istnieje dobry, krótki, zwięzły i jednoznaczny odpowiednik wyrazu obcego używajmy wyrazów polskich. Można chyba tego wymagać od Polaka, prawda? Z innych nagminnych ciekawostek można przytoczyć uporczywe pisanie assembler zamiast spolszczonej (krótszej!) wersji asembler czy używanie apostrofu (') przy odmianie angielskich wyrazów zakończonych na spółgłoskę (np. w Devpac'u zamiast w Devpacu, itp.). Zainteresowanym ciekawą pod względem języka polskiego literaturą polecam zresztą lekturę naszych pism komputerowych, niestety nie wyłączając "Amigowca".

Przepraszam tych, którzy poczują się dotknięci formą tych uwag, lecz przesiedziałem cicho prawie dwa lata i jakoś nie zauważyłem by fachowcy od języka (programowania) wykazali się jakąkolwiek kulturą językową (teraz chodzi o ten język w buzi). Myślę, że czasem warto przeczytać to, co się samemu pisze. Po prostu przez szacunek do innych i do siebie. Zwykle okazuje się, że można napisać prościej i jaśniej - zupełnie jak przy pisaniu programu. Mam jednak wrażenie, że niektórzy są uczuleni na własną prozę. Niestety

Z nadzieją, że komuś coś z tego artykułu gdzieś się zaalokowało w resetoodpornym radzie - kończę. ❑

Dźwięk i muzyka w programach AMOSA

Tomasz Hrycuniak

Ten odcinek naszego kursu programowania będzie poświęcony tworzeniu efektów dźwiękowych i stosowaniu modułów muzycznych we własnych programach. Amos pozwala w łatwy sposób wykorzystać wszystkie możliwości dźwiękowe Amigi, co ma przede wszystkim duże znaczenie dla piszących w tym języku gry i programy demonstracyjne. W tego typu programach dobry dźwięk, to co najmniej połowa sukcesu. Myślę jednak, że wszyscy z chęcią zapoznają się z metodami udźwiękowiania swoich programów.

Na początku przedstawię możliwości Amosa w dziedzinie generowania dźwięku. Będą to zarówno proste efekty dźwiękowe, jak i całkiem profesjonalne odtwarzanie z nut. Zaczynamy:

Boom

Instrukcja ta wywołuje dźwięk eksplozji. Jest stosowana zazwyczaj jako najprostszy efekt dźwiękowy w grach. Jeśli oczywiście mają one coś wspólnego z wybuchami.

Shoot

Efekt wystrzału. Zastosowanie podobne jak poprzednio. Należy jednak pamiętać, że instrukcja nie powoduje zatrzymania wykonywania programu. Jeśli więc bezpośrednio po niej, umieszczone zostaną inne instrukcje dźwiękowe, może dojść do kolizji między nimi i dźwięk jaki usłyszymy, z wystrzałem będzie miał niewiele wspólnego. W takim wypadku po komendzie Shoot umieścić trzeba instrukcję Wait. Wstrzyma ona wykonywanie następnych instrukcji na określony czas, który należy dobrać tak, aby dźwięk mógł wybrzmieć do końca (np. Wait 6).

Bell f

f - częstotliwość dźwięku

Komenda ta daje czysty dźwięk przypominający dzwon.

Play v,h,t

v - kanał
h - wysokość dźwięku
t - czas

Instrukcja ta pozwala na odtwarzanie dźwięku z nut. Za pomocą zmiennej v można ustalić kanał dźwiękowy, na którym będzie odtwarzany dźwięk. Możemy wybrać jeden z 4 kanałów jakimi dysponuje Amiga (1 i 3 - lewy kanał stereo, 2 i 4 prawy kanał stereo). Wysokość dźwięku określana jest liczbą z przedziału od 0 do 96. Wartości odpowiadające poszczególnym nutom gamy są podane w tabeli.

Parametr t określa czas jaki ma upłynąć przed wykonaniem następnej instrukcji. Pozwala to na ustalenie odpowiednich odstępów między nutami w utworze. Zastosowanie tej komendy jest bardzo szerokie. Z jej pomocą można wytwarzać zarówno dźwięki pojedyncze o charakterze sygnalizacyjnym, jak i całe melodie. Można także przyporządkować pojedyncze dźwięki wybranym klawiszom i uzyskać w ten sposób coś w rodzaju instrumentu muzycznego. Sposób wykonania może być następujący:

Listing 1

```
Led Off
Do
If Key State(80)<0 Then Play 1,37,5
If Key State(81)<0 Then Play 2,39,5
If Key State(82)<0 Then Play 3,41,5
If Key State(83)<0 Then Play 4,42,5
If Key State(84)<0 Then Play 1,44,5
If Key State(85)<0 Then Play 2,46,5
If Key State(86)<0 Then Play 3,47,5
If Key State(87)<0 Then Play 4,48,5
Loop
End
```

Programik ten pozwala nieco sobie "pograć" na klawiszach funkcyjnych (F1 do F8).

Set Wave f,k$

f - numer definiowanej fali
k$ - zmienna tekstowa, zawierająca liczby określające kształt fali

Instrukcja ta umożliwia definiowanie własnych instrumentów dla komendy Play, poprzez zdefiniowanie kształtu fali przechowywanego w pamięci i będącego wzorcem odtwarzanego dźwięku. Fale o numerach 0 i 1 są już w Amosie zdefiniowane. Fala o numerze 0 jest losowo ustalanym wzorcem dla efektu eksplozji, a 1 to czysta sinusoida będąca instrumentem podstawowym dla instrukcji Play. Zmienna tekstowa k$ musi zawierać ciąg 255 liczb określających kształt fali dźwiękowej. Oto przykładowy sposób zdefiniowania fali trójkątnej. (Patrz listing 2)

Wave f To v

f - numer zdefiniowanej fali
v - numer kanału

Listing 2

```
k$=""
For I=-128 To 127
x=I
if x<0 then Add x,256
k$=k$+Chr$(x)
Next I
Set Wave 2,k$
End
```

Komenda ta przyporządkowuje falę zdefiniowaną za pomocą instrukcji Set Wave do określonego kanału. Po wykonaniu tej komendy, wszystkie dźwięki odtwarzane przez wybrany kanał (lub kanały) będą tworzone na podstawie fali zdefiniowanej przez użytkownika.

Set Envel f,o to t,p

f - numer fali
o - faza
t - długość fazy
p - poziom głośności

Instrukcja ta ustala obwiednię dźwięku, czyli zmiany jego natężenia w czasie odtwarzania. Mamy możliwość dokonania płynnych zmian w głośności dźwięku generowanego na podstawie wybranej definicji fali. Każdy dźwięk dzielony jest na 7 części następujących kolejno po sobie w czasie odtwarzania. W każdej z nich może następować inna zmiana natężenia (zwiększanie lub zmniejszanie głośności). Zmienna o ustala, którą z faz dźwięku będziemy definiować (0 - to pierwsza faza dźwięku, 6 to ostatnia). t - to czas trwania wybranej fazy w 1/50 s, a p to głośność dźwięku jaka ma zostać osiągnięta pod koniec definiowanej fazy (0 do 63). Wartością początkową dla danej fazy jest wartość końcowa fazy poprzedniej bądź wartość maksymalna. Jako przykład zastosowania tej instrukcji proponuję modyfikację poprzedniego programiku polegającą na zastosowaniu definiowanej przez użytkownika fali dźwiękowej, złożonej z kilku sinusoid i zmienionej obwiedni dźwięku. (Patrz listing 3)

Także i tym razem można sobie "pograć" na klawiszach funkcyjnych, lecz nie będzie to już standardowy dźwięk (czysta sinusoida), a barwa definiowana przez użytkownika.

Teraz już czas zająć się wykorzystaniem gotowych dźwięków w formie sampli i modułów muzycznych. Pozwalają one na osiągnięcie znacznie ciekawszych efektów, niż przy stosowaniu dźwięków generowanych. Trzeba jednak pamiętać o sporej ilości pamięci

Listing 3

```
Led Off
Degree
S$=""
For S=0 To 255
V=Int((Sin(S)/2+Sin
(S*2+45)/4)*128)+127
S$=S$+Chr$(V)
Next S
Set Wave 2,S$
Wave 2 To 15
Set Envel 2,0 To 15,60
Set Envel 2,1 To 1,50
Set Envel 2,2 To 10,50
Set Envel 2,3 To 50,0
Key Speed 20,10
Do
If Key State(80)<0 Then Play 1,37,10
If Key State(81)<0 Then Play 2,39,10
If Key State(82)<0 Then Play 3,41,10
If Key State(83)<0 Then Play 4,42,10
If Key State(84)<0 Then Play 1,44,10
If Key State(85)<0 Then Play 2,46,10
If Key State(86)<0 Then Play 3,47,10
If Key State(87)<0 Then Play 4,48,10
Loop
End
```

potrzebnej do przechowywania spróbkownego dźwięku. Rozpocznę od kilku komend pozwalających na wykorzystanie pojedynczych sampli.

Sample n To v

n - numer sampla
v - numer kanału

Instrukcja ta przyporządkowuje wybranemu kanałowi sampla o podanym numerze. Tym samym wybrany fragment spróbkowanego dźwięku stanie się instrumentem dla instrukcji Play. Trzeba pamiętać, że sample z których chcemy korzystać, muszą znajdować się w odpowiednim banku pamięci (standardowo jest to bank nr 5). Kilka takich banków nagranych jest na dysku z Amosem i można je wczytać za pomocą instrukcji Load. Jeśli jednak ktoś zechce użyć własnych sampli, to do umieszczenia ich w banku pamięci najwygodniej będzie zastosować program Sample_Bank_Maker, który także powinien znajdować się na dysku z Amosem.

Sam Play v,s,f

v - numer kanału
s - numer sampla
f - częstotliwość

Instrukcja ta pozwala na odtworzenie przez kanał v sampla o numerze s. Zmienna f określa częstotliwość odtwarzania w Hercach. Ustala ona ilość próbek jakie mają być odtworzone w czasie jednej sekundy. Typowe wartości leżą w przedziale od 4000 do 10000, przy czym odpowiednio manipulując tą wielkością, można z jednego sampla przechowywanego w pamięci uzyskać kilka różnie brzmiących dźwięków.

Sam Bank n

n - numer banku

Zmiana numeru banku pamięci przeznaczonego do przechowywania sampli. Pozwala to na umieszczenie w pamięci kilku zestawów sampli i następnie wybieranie, z którego dane mają być pobierane.

Tabela 1. Wartości poszczególnych nut gamy.

Dźwięk	Oktawa							
	1	2	3	4	5	6	7	
	Wysokość							
C	1	13	25	37	49	61	73	85
Cis	1	2	14	26	38	50	62	74
D	3	15	27	39	51	63	75	87
Dis	4	16	28	40	52	64	76	88
E	5	17	29	41	53	65	77	89
F	6	18	30	42	54	66	78	90
Fis	7	19	31	43	55	67	79	91
G	8	20	32	44	56	68	80	92
Gis	9	21	33	45	57	69	81	93
A	10	22	34	46	58	70	82	94
B	11	23	35	47	59	71	83	95
H	12	24	36	48	60	72	84	96

Sam Loop On/Off

Włączanie lub wyłączanie "zapętlenia" wszystkich sampli. Włączenie tej opcji powoduje, że wszystkie odtwarzane materiały będą nieprzerwanie powtarzane.

Zanim przedstawię instrukcje pozwalające na odtwarzanie modułów muzycznych, koniecznych jest kilka wyjaśnień. Amos może bezpośrednio odtwarzać tylko moduły zapisane w formie banku pamięci (rozszerzenie .abk). Niezbędne jest więc wykonanie konwersji modułów zapisanych w innych formatach. W tym celu na dysku z Amosem zostały umieszczone programy konwertujące na odpowiedni standard. Dla najpopularniejszych modułów w standardzie Noisetrackera i Protrackera jest to program Sound_tracker2_1.AMOS. Uwaga!!! Nie dotyczy to pakietu Amos Professional, który zawiera komendy pozwalające na bezpośrednie odtwarzanie modułów w kilku standardach. Wspomnianymi instrukcjami, a także innymi nowymi komendami pakietu Amos Pro zajmę się w jednym z następnych odcinków.

Music n

n - numer utworu

Instrukcja ta rozpoczyna odtwarzanie utworu o numerze n z muzycznego banku pamięci (nr 3), o ile taki został utworzony. Bank z muzyką (np. utworzony programem konwertującym) można wczytać z dyskietki tak jak każdy inny bank pamięci - instrukcja Load "tytuł". Uwaga!!! Banki z programu konwertującego zawierają tylko jeden utwór dostępny pod numerem 1. Rozpoczęcie odtwarzania muzyki oczywiście nie wstrzymuje wykonywania programu i zaraz po komendzie Music mogą następować kolejne instrukcje przeznaczone do wykonania. Muzyka będzie więc stanowić swego rodzaju tło dla napisanego przez nas programu, np. dema.

Music Off

Wyłącza odtwarzanie muzyki. Aby ponownie ją uruchomić, należy jeszcze raz użyć komendy Music.

Tempo p

p - prędkość odtwarzania

Instrukcja ta zmienia tempo odtwarzania utworu. Zmienna p może przyjmować wartości od 1 do 100.

Mvolume n

n - natężenie dźwięku

Komenda ta ustala natężenie dźwięku w zakresie od 0 do 63.

Na zakończenie zaprezentować wszystkim nieco dłuższy program pokazujący zastosowanie omówionych wcześniej komend. Jest to Player odtwarzający moduły muzyczne zapisane w formacie Amosa (rozszerzenie .abk). Możliwe jest wczytanie z dysku dowolnego utworu, jego odtworzenie oraz regulacja głośności i tempa, przez kliknięcie myszką w odpowiednim polu (w czasie odtwarzania utworu). Wartość ustawionej głośności i tempa jest obrazowana za pomocą dwóch "słupków" zmieniających swą długość w czasie regulacji. Program jest napisany przejrzyście i z pewnością nie będzie problemów z przeanalizowaniem jego działania. (Patrz listing 4)

Listing 4

```
Rem Odtwarzanie danych muzycznych
w formacie AMOS'a
Rem
Led Off
Curs Off
Flash Off
Palette ,$0
F$=Fsel$("*.abk","","Wybierz plik")
Load F$
T=17
V=63
Locate 1,10
Print "Tempo"
Locate 1,15
Print "Głośność"
Locate 1,1
Print "Głośność"
Locate 12,1
Print "Tempo"
Locate 23,3
Print "Start"
Locate 30,3
Print "Stop"
Locate 3,3
Print "+"
Locate 6,3
Print "-"
Locate 13,3
Print "+"
Locate 16,3
Print "-"
Box 20,20 To 35,35
Box 45,20 To 60,35
Box 100,20 To 115,35
Box 125,20 To 140,35
Box 180,20 To 227,35
Box 233,20 To 280,35
Bar 10,130 To V*4.76+10,140
Bar 10,90 To T*3+10,100
Reserve Zone 6
Set Zone 1,20,20 To 35,35
Set Zone 2,45,20 To 60,35
Set Zone 3,100,20 To 115,35
Set Zone 4,125,20 To 140,35
Set Zone 5,180,20 To 227,35
Set Zone 6,233,20 To 280,35
Do
While Mouse Zone=1
If Mouse Key=1
If V<63 : V=V+1 : End If
Wait 3 : Mvolume V
Bar 10,130 To V*4.76+10,140
End If
Wend
While Mouse Zone=2
If Mouse Key=1
If V>0 : V=V-1 : End If
Wait 3 : Mvolume V
Home
If V>0
Ink 1
Bar V*4.76+10,130 To V*4.76+15,140
Ink 2
End If
End If
Wend
While Mouse Zone=3
If Mouse Key=1
If T<100 : T=T+1 : End If
Wait 3 : Tempo T
Bar 10,90 To T*3+10,100
End If
Wend
While Mouse Zone=4
If Mouse Key=1
If T>1 : T=T-1 : End If
Wait 3 : Tempo T
If T>0
Ink 1
Bar T*3+10,90 To T*3+13,100
Ink 2
End If
End If
Wend
While Mouse Zone=5
If Mouse Key=1 Then Music 1 : Tempo T
Wend
While Mouse Zone=6
If Mouse Key=1 Then Music Off
Wend
Loop
End
```

Myślę też, że programik ten będzie przydatny wszystkim zajmującym się programowaniem w Amosie, ponieważ nie spotkałem jeszcze playera rozpoznającego standard muzycznych banków pamięci Amosa. Jak zwykle na koniec, życzę wszystkim czytelnikom AMIGOWCA powodzenia i zachęcam do lektury następnego odcinka, który będzie poświęcony "grafice ruchomej". Opiszę między innymi zasady tworzenia i używania bobów oraz spriteów. ❑

AREXX cz.3

Izabela Skibińska

W tym odcinku zajmiemy się tworzeniem programów z interfejsem Arexxa. Oprócz tych praktycznych umiejętności, które pozwolą, aby nasze aplikacje były zgodne ze standardem zalecanym przez firmę Commodore, będziemy mieli możliwość lepiej poznać zasady działania Arexxa w systemie operacyjnym Amigi.

Listing 1 obrazuje najprostszy wariant realizacji programu z portem Arexxa. Niezbędne funkcje i struktury do realizacji naszego celu znajdują się w bibliotece rexxsyslib.library. Przykładowy program posiada jedynie możliwość odbierania arexxowych komunikatów, które możemy wysyłać za pomocą skryptu send.rexx. Aby odbierać te rozkazy należy stworzyć message port (pomocne tu będą funkcje CreatePort z amiga.lib bądź CreateMsgPort z exec.library dla użytkowników Kickstartu 2.0 wzwyż) o unikalnej nazwie. W naszym przypadku nazwaliśmy go "OURPORT". Dla systemu operacyjnego Amigi istotne są duże i małe litery w nazwie portu, przy czym jest brane pod uwagę jedynie 16 pierwszych znaków. Po stworzeniu portu sytuacja jest już bardzo podobna jak w standardowym programie badającym nadchodzące komunikaty z Intuition, z tym, że w naszym przypadku chodzi oczywiście o struktury specjalnie przystosowane do przesyłania rozkazów arexxowych (RexxMsg).

W myśl obiektowej konstrukcji całego systemu operacyjnego Amigi struktury te zawierają w sobie podstawową strukturę Message z exec.library oraz dodatkowe pola dla użytku Arexxa, którymi to zajmiemy się w naszym przykładzie. Aby analizować jedynie to, o co nam chodzi (komunikaty Arexxa) korzystamy z funkcji IsRexxMsg z rexxsyslib.library. Zestaw funkcji tej biblioteki znajduje się w tabelce na końcu artykułu.

Adres łańcucha reprezentującego rozkaz Arexxa pobieramy z pola rm_Args[0]. Następnie badamy czy nasz program rozumie polecenie. Gdy rozkaz jest znany wpisujemy odpowiedni kod powrotu oraz wynik w pola rm_Result1 i rm_Result2. Ponieważ to do naszego programu został wysłany komunikat, konieczne jest wywołanie ReplyMsg() i to niezależnie od tego czy rozkaz jest poprawny czy też nie.

Zwróćcie też uwagę, że skorzystaliśmy z wariantu funkcji (strnicmp) porównującej ciągi znakowe, nie rozróżniającej dużych i małych liter. Uczyniliśmy to w myśl ogólnej zasady pisania rozkazów Arexxa zarówno dużymi, jak i małymi literami.

Jak natomiast wygląda to wszystko od strony skryptu arexxowego? Rozkazem ADDRESS "portname" ustalany jest adres "hostu", czyli programu posiadającego port Arexxa o nazwie portname. Tam zostaną skierowane wszystkie te rozkazy, których nie rozumie interpreter. W poprzednim odcinku używaliśmy ADDRESS COMMAND, gdy chcieliśmy uzyskać komunikację z AmigaDosem. Jest to wyjątek od obowiązku podawania nazwy portu.

Komenda OPTIONS RESULTS powoduje, że wynik działania rozkazu znajdzie się w zmiennej RESULT. W praktyce oznacza to, że komunikat wysłany do naszego "hostu" ma ustawiony bit RXFB_RESULT w polu rm_Action. Tak więc program musi odpowiednio potraktować otrzymaną wiadomość. W przypadku pomyślnego rozkazu ustawiamy kod powrotu w polu rm_Result1 na 0 (stała RC_OK). W polu rm_Result2 umieszczamy adres tzw. Argstringu, czyli łańcucha wynikowego, który to tworzymy funkcją CreateArgstring. Jeden z parametrów funkcji to długość łańcucha znaków. Jest to konieczne, gdyż pamięć na ciąg znakowy nie zostaje zazwyczaj zwolniona przez program, który ją zarezerwował.

W zmiennej RC jest umieszczone to, co znajduje się w polu rm_Result1. Wybór wartości kodu powrotu zależy więc od projektującego aplikację z interfejsem Arexxa. Zalecam stosowanie się do następującego standardu:

1-9 - ostrzeżenia i mniej znaczące błędy
10-19 - "umiarkowane błędy"
20 + - poważne błędy

Zgodnie z zaleceniami nie należy również zmieniać innych pól instancji struktury RexxMsg niż rm_Result1 i rm_Result2. W naszym przypadku umieściliśmy kod błędu o numerze 10 (predefiniowana zmienna RC_ERROR) w przypadku podania błędnej liczby argumentów dla rozkazów ADD i SUB.

Gdy wiemy już jak przyjmować komunikaty Arexxa możemy zająć się wysyłaniem takowych. Wyobraźmy sobie, że piszemy program, w którym chcemy badać wcześniej już uruchomionym programem VirusChecker, czy dyski w stacji df0: wolne są od wirusów.

Cały problem rozwiąże nam procedura sendToVirusChecker, której parametrami są w kolejności: wskaźnik na nasz port, jego nazwa, komenda dla Virus Checkera oraz maska sygnałów naszego portu. Tworzeniem komunikatów arexxowych (RexxMsg) zajmuje się funkcja z rexxsyslib.library CreateRexxMsg. Istotnymi wartościami dla struktury RexxMsg w momencie jej tworzenia jest wskaźnik i nazwa portu, gdzie

```
/* Listing1 */

 #include <exec/types.h>
 #include <exec/ports.h>
 #include <dos/dos.h>
 #include <clib/alib_protos.h>
 #include <clib/exec_protos.h>
 #include <clib/rexxsyslib_protos.h>
 #include <pragmas/rexxsyslib_pragmas.h>
 #include <rexx/storage.h>
 #include <rexx/errors.h>
 #include <string.h>
 #include <stdlib.h>

 struct RxLib *RexxSysBase;

 VOID handleArexxMessage(struct RexxMsg *);
 WORD (*functions[2])();
 char *names[2] = {"add","sub"};

 WORD add(WORD a1,WORD a2)
 {
 return(a1+a2);
 }

 WORD sub(WORD a1,WORD a2)
 {
 return(a2-a1);
 }

 main()
 {
   struct MsgPort *Port;
   ULONG arexportsig,usersig,signal;
   struct RexxMsg *ArexxMsg;
   BOOL koniec=FALSE;

  if (!(RexxSysBase = (struct RxLib *)OpenLibrary((UBYTE *)"
  rexxsyslib.library",0L)))
     exit(1L);

  if (!(Port = CreatePort("OURPORT",0)))
  {
     CloseLibrary((struct Library *) RexxSysBase);exit(1L);
  }

  arexportsig = 1L << Port->mp_SigBit;
  usersig = SIGBREAKF_CTRL_C;
  functions[0] = add;
  functions[1] = sub;

  puts("Press CTRL-C to close the port and stop the program");

  while (koniec == FALSE)
  {
    signal = Wait(arexportsig | usersig);
    if (signal & arexportsig)
    {
      while (ArexxMsg = (struct RexxMsg *)GetMsg(Port))
        if (IsRexxMsg(ArexxMsg))
        handleArexxMessage(ArexxMsg);
    }
   if (signal & usersig)
   {
    koniec = TRUE;
    DeletePort(Port);
    CloseLibrary((struct Library *)RexxSysBase);
    puts("Port removed ");
   }
  }
 }

  void handleArexxMessage(struct RexxMsg *mess)
                {
     WORD i, k, arg1, arg2, ret, len;
     UBYTE nazwa[5], *command = (char *)mess->rm_Args[0], buf[7];

     mess->rm_Result1 = RC_ERROR     ; /* Gdy rozkaz nie znany lub
                                           liczba argumentów zła */
     mess->rm_Result2 = 0L;

     for(i=0;i<3;i++)
     {
       if (strnicmp(names[i],command,strlen(names[i]))==0)
       {
         k = sscanf(command,"%s %hd %hd",nazwa,&arg1,&arg2);
         if (k == 3)
         {
           ret = (*functions[i])(arg1,arg2);
           len =  stci_d(buf,ret);  /*konwersja na string */
           mess->rm_Result1 = RC_OK;
           mess->rm_Result2 = 0L;
           if (mess->rm_Action & (1L << RXFB_RESULT))
           mess->rm_Result2 = (LONG)CreateArgstring(buf,len);
         }
       break;
       }
    }
    ReplyMsg((struct Message *)mess);
 }

  /*send.rexx */

   address "OURPORT"
   options results

   add 12 78
   say result
   add 15
   say rc
   sub 13 90
   say result
```

Uwaga! Tekst wytłuszczony należy wpisywać w jednej linii.

ma być odsyłana po wywołaniu przez odbiorcę funkcji odpowiedź (ReplyMsg). W naszym przypadku jest to naturalnie port naszego programu. Pozostaje nam jedynie zdefiniowanie typu żądanej operacji. Skoro ma być to komenda, ustawiamy pole rm_Action na wartość stałej RXCOMM. Komunikat potraktowany funkcją ReplyMsg poznajemy po ustawieniu flagi NT_REPLYMSG. Tak skonstruowana funkcja sendToVirusChecker jest odpowiedzialna zarówno za tworzenie, jak i odbieranie stworzonych przez siebie struktur RexxMsg.

Ponieważ wiemy jak odróżnić "odpowiedź" od komunikatu wysłanego np. ze skryptu send.rexx, równie dobrze moglibyśmy rozpatrywać wszystkie komunikaty przychodzące do naszego portu funkcją handleArexxMessage. Na uwagę zasługuje fakt konieczności wyłączenia multitaskingu w momencie wysyłania komunikatów do określonego portu (funkcja PutMsg), aby uniemożliwić jego skasowanie w tym czasie. Przedstawiony wyżej mechanizm umożliwia nam wysyłanie pojedynczych rozkazów do programów z portem Arexxa.

Co jednak zrobić, gdy potrzebne jest bardziej skomplikowane postępowanie decyzyjne, jednym słowem program? Nic prostszego. Elastyczność Arexxa pozwala na zastosowanie identycznego mechanizmu z tym, że zamiast rozkazu w polu rm_Args[0] musi znaleźć się nazwa programu. Taki komunikat będący normalnym skryptem arexxowym wysyłamy już naturalnie do portu interpretera "REXX", aby ten przeanalizował program.

Bardzo pomocną sprawą jest tu tzw. sufiks (rozszerzenie) będący drugim parametrem funkcji CreateRexxMsg. Interpreter Arexxa będzie szukał skryptu z takim właśnie rozszerzeniem. Wyżej przedstawiona metoda jest zasadniczo wykorzystywana w wielu aplikacjach, gdzie klawiszami funkcyjnymi wywołujemy skrypty Arexxa. Standardowym rozszerzeniem jest ".rexx". Z reguły aplikacje stosują jednak dla przejrzystości określone rozszerzenia. Na przykład program ADPro, którym zajmiemy się w następnym odcinku, oznacza swoje programy arexxowe rozszerzeniem "adpro". Wróćmy do naszego przykładu. Aby wykonywał on tę samą operację, ale za pośrednictwem skryptu arexxowego wystarczy nazwę pliku z następującą zawartością:

```
/*Comment*/
ADDRESS 'Virus_Checker'
'checkdrive\df0:'
```

umieścić w polu rm_Args[0], a następnie zmienić parametr wywołania FindPort na "REXX".

```
/* Listing 2 */

/* procedura sendToVirusChecker */
/* wywołanie: sendToVirusChecker(Port,"OURPORT",
command,arexportsig); */
/* gdzie command np "checkdrive\df0:" (składnia komend Arexxa Virus */
/* Checkera wymaga aby argumenty były odzielone od komendy
znakiem "" )*/

  BOOL sendToVirusChecker(struct MsgPort *Port, UBYTE *Name,
                          UBYTE *Command, ULONG ourportsig)
    {

      struct RexxMsg *AMsg;
      struct MsgPort *VCPort;

      if (!(AMsg = (struct RexxMsg *) CreateRexxMsg(Port,
      NULL,Name)))
        return(FALSE);

      if(!(AMsg->rm_Args[0] = CreateArgstring(Command,
      (LONG )strlen(Command))))
        return(FALSE);
      AMsg->rm_Action = RXCOMM;
      if (VCPort = FindPort("Virus_Checker"))
      {
        Forbid();
        PutMsg(VCPort,(struct Message *)AMsg);
        Permit();
      }
      else
      {
        DeleteArgstring(AMsg->rm_Args[0]);
        DeleteRexxMsg(AMsg);
      }

      Wait(ourportsig);
      while(AMsg = (struct RexxMsg *)GetMsg(Port))
      {
        if (AMsg->rm_Node.mn_Node.ln_Type == NT_REPLYMSG)
          {
            if (AMsg->rm_Result1 == RC_OK)
            DeleteArgstring(AMsg->rm_Args[0]);
            DeleteRexxMsg(AMsg);

          }
      }
    }
```

Uwaga! Tekst wytłuszczony należy wpisywać w jednej linii.

Funkcje z biblioteki "rexxsyslib.library":

ClearRexxMsg(struct RexxMsg *RexxMsg,ULONG count)
a0 d0
RexxMsg - wskaźnik na RexxMsg
count - liczba pól do wyzerowania
Zwalnia wszystkie "Argstringi" i zeruje 16 pól struktury RexxMsg(rm_Args) uprzednio na nie wskazujące.

UBYTE *CreateArgstring(UBYTE *str,ULONG length)
a0,d0 a0 d0
str - wkaźnik na ciąg wejściowy
length - długość ciągu wejściowego
Przydziela strukturę RexxArg. Rezultatem funkcji jest wskaźnik na Argstring.
(pole w strukturze)

struct RexxMsg *CreateRexxMsg(struct MsgPort *port,UBYTE *ext,UBYTE *host)
d0,a0 a0 a1 a2
port - wskaźnik portu, gdzie będzie zwracany komunikat (funkcją ReplyMsg())
ext - wskaźnik rozszerzenia (sufiksu) będącego standardowym rozszerzeniem skryptów arexxowych
host - wskaźnik nazwy standardowego "host" portu.
Gdy NULL, portem tym staje się REXX. Zwraca wskaźnik na utworzoną przez siebie strukturę RexxMsg.

VOID DeleteArgstring(UBYTE *argstring)
a0
argstring - wskaźnik na Argstring
Zwalnia pamięć na Argstring zarezerwowaną przez CreateArgstring().

VOID DeleteRexxMsg(struct RexxMsg *packet)
a0
packet - wskaźnik na strukurę RexxMsg
Zwalnia pamięć na RexxMsg, zarezerwowaną przez CreateRexxMsg.

BOOL FillRexxMsg(struct RexxMsg *packet,ULONG count,ULONG mask)
D0 A0 D0 D1
packet - j.w.
count - liczba pól rm_Args (w strukturze RexxMsg) do wypełnienia
mask - maska bitowa służąca do określenia każdego z 16 pól rm_Args
Inicjuje do 16 pól w strukturze RexxMsg wskaźnikami na Argstringi lub liczbami, co jest określone maską. Gdy bit 1 - wskaźnik na Argstring, 0 - liczba.

BOOL IsRexxMsg(struct RexxMsg *packet)
D0 A0
packet - j.w
Zwraca TRUE, gdy packet wskazuje na RexxMsg, FALSE w p.p.

ULONG LengthArgString(UBYTE *argstring)
D0 A0
argstring - wskaźnik na Argstring
Zwraca długość Argstringu.

VOID LockRexxBase(ULONG resource)
D0
resource - stała definiująca, co konkretnie uchronić
Na razie musi być ZERO (czyli wszystko). Chroni przed wielodostępem strukturę opisującą rexxsyslib.library.

VOID UnLockRexxBase(ULONG resource)
D0
Odwrotne działanie do poprzedniej funkcji. ❑

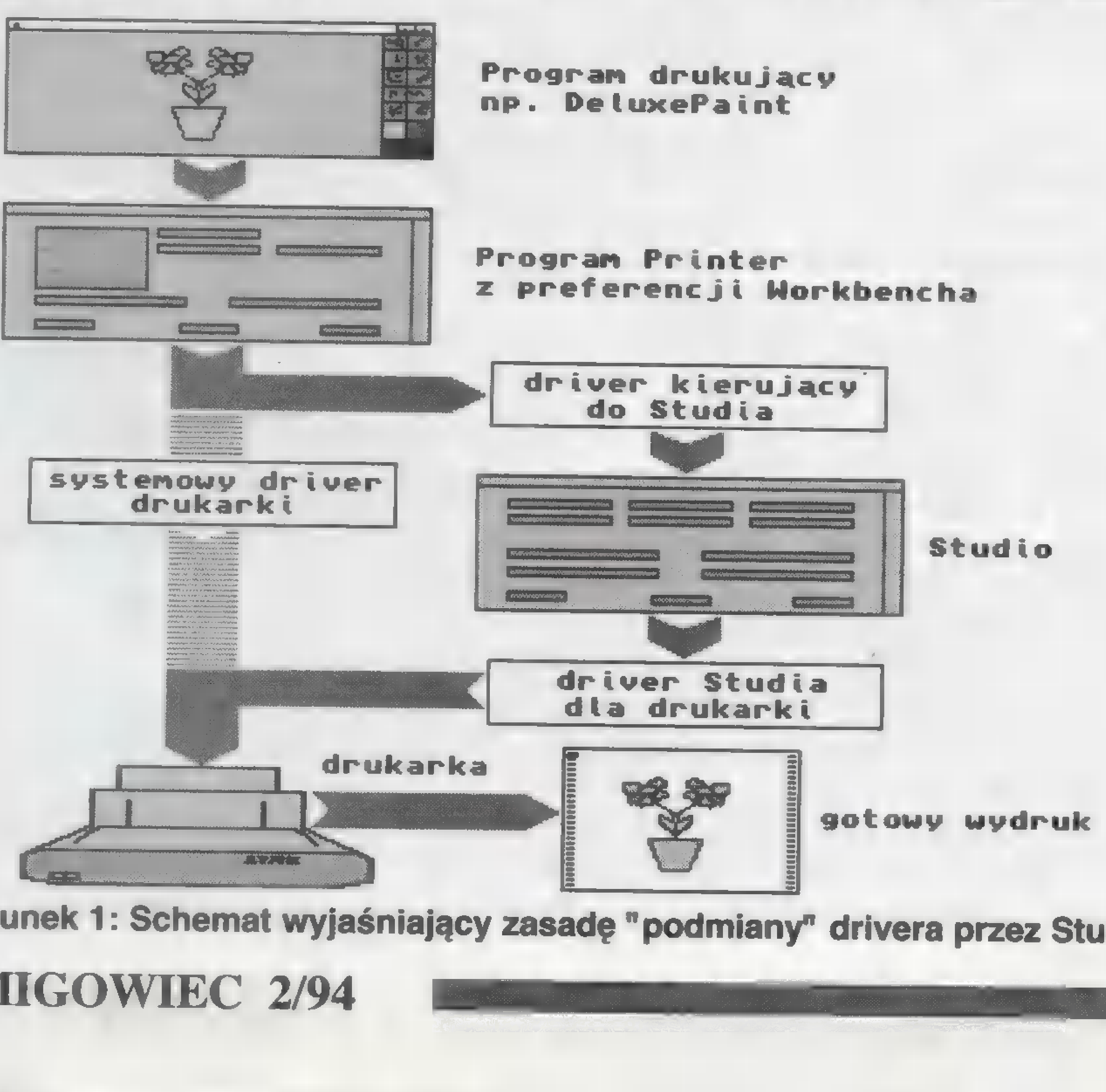
STUDIO

Maciej Klimkiewicz

Studio jest programem wspomagającym obsługę drukarek igłowych, laserowych i atramentowych. Przeznaczony jest przede wszystkim dla tych użytkowników, którzy nie są zadowoleni z wydruków otrzymywanych przy pomocy driverów oferowanych przez system.

Amiga na całym świecie znana jest ze swoich możliwości graficznych. Duże rozdzielczości rysunków dochodzące nawet do kilku tysięcy pikseli oraz ogromna paleta kolorów, zadecydowały o sukcesie tego komputera na rynku graficznym. Uzyskiwane obrazki mogą być bez problemu wyświetlane przez monitory. Problem powstaje, gdy należy je przenieść na kartkę papieru. Większość drukarek oferuje tylko czarno-biały wydruk. Co prawda przy pomocy triku polegającego na stosowaniu rastrów, możliwe jest uzyskanie wielu odcieni szarości, jednak barierą stojącą na przeszkodzie jest ograniczona rozdzielczość druku. Szczególnie dotkliwie odczuwają to właściciele drukarek igłowych i atramentowych, dla których drivery dostarczone wraz z Workbenchem często nie wykorzystują nawet połowy dostępnych możliwości. W przełamaniu tych ograniczeń ma pomóc specjanie do tego celu stworzone Studio.

Studio dostarczane jest na jednej dyskietce, na której znajduje się program instalacyjny wraz z nowymi driverami. Do działania wymagane jest minimum 512 KB pamięci RAM oraz system nie starszy niż 1.3.

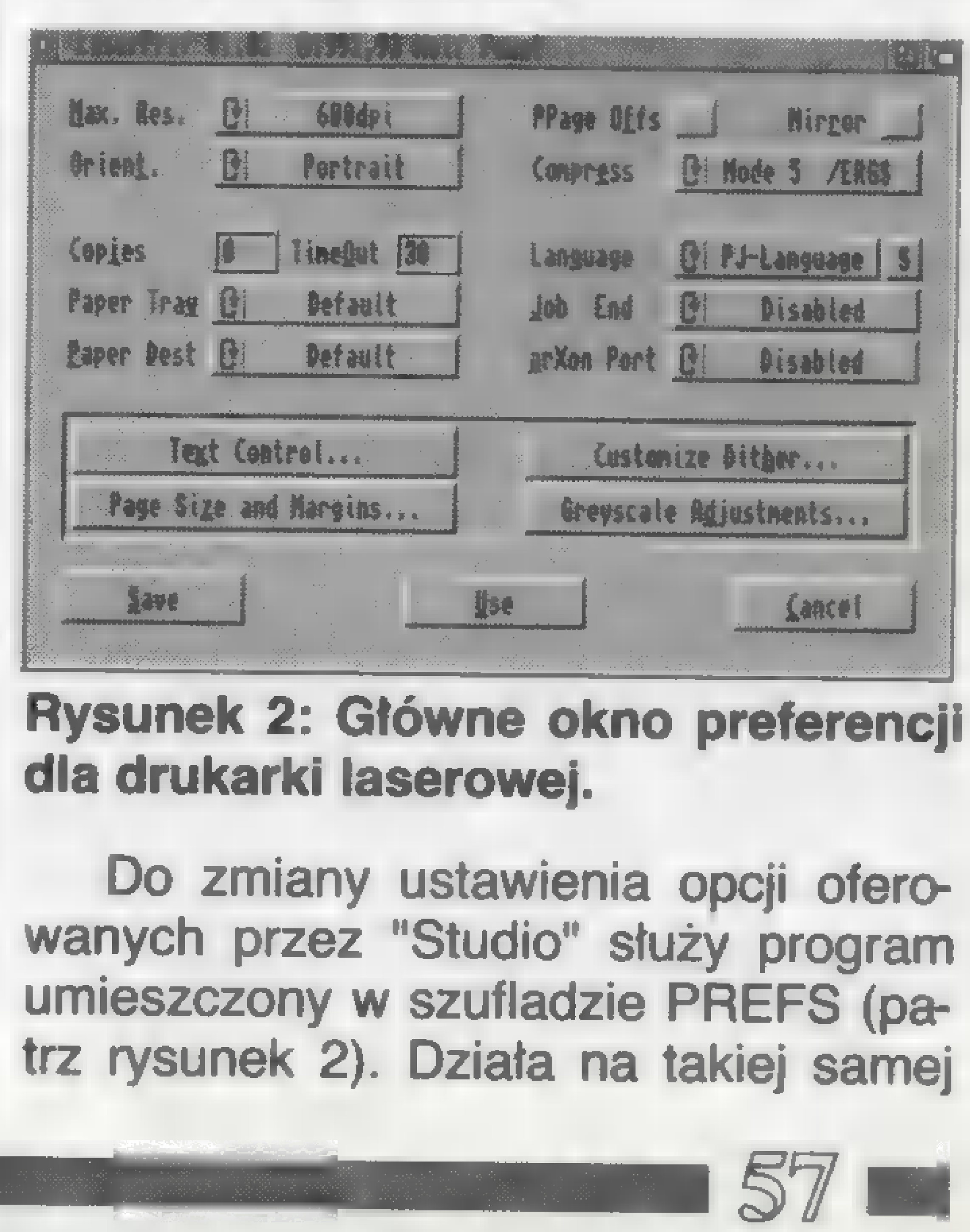


Rysunek 1: Schemat wyjaśniający zasadę "podmiany" drivera przez Studio.

Już na wstępie podczas instalowania należy podać posiadaną drukarkę. Oferowane są następujące grupy:

- 24 i 48 igłowe drukarki emulujące "Epsona", "Stara", "Neca" lub inne znane tryby,
- drukarki kolorowe HP DeskJet starszych i nowszych typów,
- drukarki laserowe HP LaserJet łącznie z ostatnią, najnowszą HP LaserJetIV.

Podczas instalacji tworzony jest własny mikrosystem obsługi drukarki. Polega on na tym, że do szuflady DEVS: PRINTERS ładowany jest specjalny driver do którego odwołują się programy w czasie drukowania. Trzeba go tylko wybrać w PREFS/PRINTER. Konstrukcja drivera sprawia, że informacje zamiast na złącze drukarki przesyłane są do "Studia", który następnie po swojemu je obrabia. Schemat tego działania przedstawia rysunek 1. Dzięki takiemu rozwiązaniu użytkownik może w każdej chwili powrócić do systemowego rozwiązania drukowania przez wybór starego drivera.



Rysunek 2: Główne okno preferencji dla drukarki laserowej.

Do zmiany ustawienia opcji oferowanych przez "Studio" służy program umieszczony w szufladzie PREFS (patrz rysunek 2). Działa na takiej samej

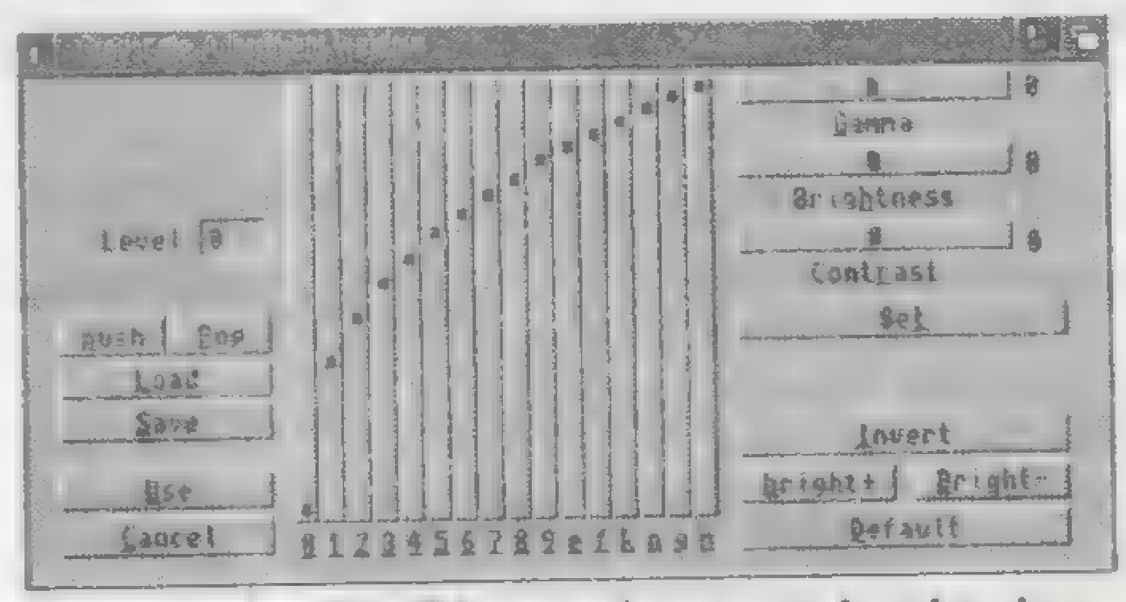
zasadzie jak stare preferencje, czyli jest "widoczny" przez wszystkie drukujące programy. Jeżeli jednak komuś nie wystarcza takie rozwiązanie, może skorzystać ze specjalnego programiku drukującego "Studio" (patrz Rysunek 4).

Podstawową obsługę drukarki w trybie tekstowym i graficznym zapewnia driver obsługiwany przez program znajdujący się w szufladzie preferencji. Ze względu na to, że dla różnych typów drukarek oferowane przez niego możliwości są odmienne, ograniczę się jedynie do podania podstawowych wspólnych cech.

W trybie tekstowym można zmieniać takie parametry jak: szerokość wszystkich czterech marginesów, wielkość odstępów między liniami, ilość znaków w linii, rozmieszczenie tabulatorów, rodzaj czcionki do drukowania itd. Poza tym dla drukarek igłowych istnieje możliwość ustalenia rodzaju trybu graficznego i tekstowego w zależności od tego, który z nich jest emulowany przez posiadany sprzęt. Jeżeli wydruk w kolorach nie jest możliwy, można je odseparować i wydrukować każdy osobno. Zakończenie drukowania może być zakomunikowane w postaci błyśnięcia ekranu albo wydania dźwięku. Możliwa jest zmiana czasu oczekiwania komputera na wydruk, po której wystąpi komunikat o braku papieru albo nieobecności drukarki. Jeżeli chodzi o drukowanie tekstów, to by było na tyle.

"Studio" swoją potęgę zawdzięcza bardzo rozbudowanemu trybowi graficznemu. Bardzo użyteczna jest możliwość określenia stopnia kompresji albo optymalizacji drukowanego obrazka. Oczywiście ma to wpływ zarówno na jakość, jak i na szybkość wydruku. Niestety obydwa parametry nie idą ze sobą w parze - im większa szybkość tym gorsza jakość i odwrotnie. Nie należy się jednak tym martwić. Algorytmy przeliczające obrazek zostały tak skonstuowane, że nawet przy względnie dużej szybkości wydruku można uzyskać doskonałe odwzorowanie obrazka na papierze. Poza tym, wedle życzenia użytkownika ruch głowicy może odbywać się w jedną, albo w obie strony. Wybór odpowiedniego trybu można pozostawić komputerowi, który sam dobierze optymalne rozwiązanie w zależności od sytuacji.

Przez zmianę stopnia zaczernienia można wpływać na zmiejszenie lub zwiększenie zużycia barwnika. Dodatkowo w przypadku drukarek atramentowych takie rozwiązanie pozwala na zlikwidowanie marszczenia się papieru od mokrego atramentu. Istnieje jeszcze jeden sposób pozwalający na zaoszczędzenie zużycia taśmy, a zarazem dodający wydrukowi uroku. Polega on na drukowaniu obrazka z rastrem. Użytkownik ma do wyboru kilka gotowych wzorów z których może skorzystać.

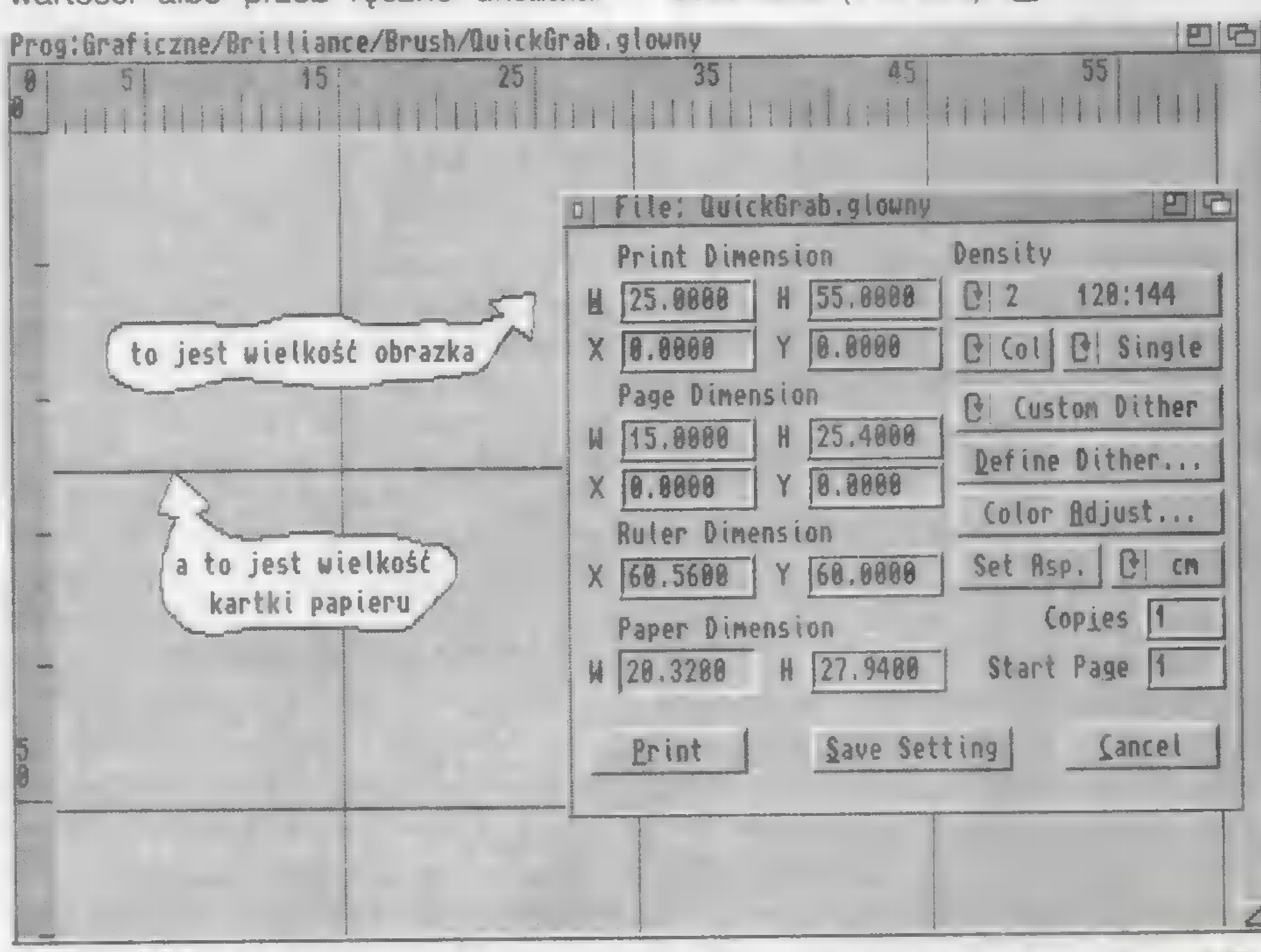


Rysunek 3: Okno do ustalania jasności, kontrastu, nasycenia, itp.

Podczas drukowania przy pomocy systemowych preferencji, często zdarza się, że otrzymany wydruk jest za jasny albo za ciemny, bądź nie posiada wymaganego kontrastu. W przypadku pracy ze "Studiem" można w znacznym stopniu zlikwidować tę wadę. Realizowane to jest przy pomocy specjalnego okna (Rysunek 3), w którym w prosty sposób określa się parametry kontrastu i jaskrawości obrazka oraz współczynnik Gamma. Ustawienie żądanych parametrów odbywa się w sposób automatyczny po wpisaniu odpowiednich wartości albo przez ręczne ukształtowanie wykresu. Żeby przekonać się o funkcjonalności takiego rozwiązania wystaczy trochę poeksperymentować.

Nie wszystkie programy graficzne pozwalają na drukowanie rysunków zgodnie z parametrami ustawionymi w preferencjach. W przypadku drukarek igłowych często zdarza się, że program poprawnie drukuje tylko w osi poziomej. Nie jest to winą drivera, ale programu z którego następuje wydruk. Po prostu część oprogramowania nie potrafi wykorzystać wszystkich informacji zawartych w preferencjach. Wyjściem z tej sytuacji jest dodatkowe narzędzie "Studia" (patrz Rysunek 4). Można w nim dokładnie ustawić szerokość i wysokość obrazka do wydruku oraz jego ułożenie względem strony. Ciekawe jest to, że ostateczny wydruk może być większy niż umożliwia to drukarka. Zrealizowano to w ten sposób, że odczytany obrazek jest dzielony na części, które drukowane są oddzielnie i które należy później posklejać.

Podsumowując, "Studio" jest doskonałym narzędziem pozwalającym w pełni wykorzystać możliwości drukarki. Jego dużą zaletą jest to, że posiada wiele różnych driverów nawet do mało znanych drukarek. Poza tym swoimi możliwościami wyprzedza wiele innych tego typu rozwiązań. Jest doskonałym następcą TurboPrinta. Jego jedyną wadą - jak na Polskie warunki, jest wysoka cena (110 DM). ❑



Rysunek 4: Dodatkowy program umożliwiający drukowanie obrazków w kilku częściach.

DROGA Z GAZETAMI

Tomasz Kokuszczyński

Wielu z Was chciałoby wykorzystywać swoje wspaniałe Amigi 500 do tworzenia gazetek szkolnych, biuletynów, czy też listów do Cioci, na dzień Dziadka. Nie wszyscy wiedzą jak to zrobić. Tym bardziej, że co nowocześniejsze programy do obróbki Danych Typowo Poczytnych (tzw. DTP) zajmują masę miejsca na dyskach twardych i wymagają niesamowitych kompleksów układów scalonych zestawionych w dużą pamięć RAM. Dzisiejsze Warsztaty DTP poświęcamy dla takich właśnie użytkowników.

Sprzęcik jaki jest...

Na początek zajmijmy się króciutkim opisem sprzętu niezbędnego do prowadzenia twórczości w zakresie DTP. Zakładam, że każdy kto chce coś zrobić w dziedzinie DTP (Desktop Publishing) musi być wyposażony przynajmniej w Amigę z 1MB pamięci i drukarkę (może być u kolegi). Jeszcze lepiej, gdy mamy 2 MB pamięci w tym 1MB Chipu oraz drugą stację. Przy większych rozdzielczościach, jakie stosuje się w DTP, da Wam się niestety we znaki nieprzyjemne drganie na monitorze. W tym wypadku można oczywiście zakupić specjalną przystawkę do Amigi określaną jako FlickerFixer, która wytłumi te drgania. Tańszym i chyba w sumie bardziej praktycznym rozwiązaniem jest poszukanie jakiegoś starego monitora monochromatycznego o przedłużonej luminescencji. Taki monitor powinien mieć zwykłe wejście MONO, a jego podłączenie do Amigi nie powinno stanowić żadnego problemu.

Twardy dysk oczywiście zawsze zwiększy szybkość Waszej pracy. Dzięki niemu można korzystać nie tylko z najprostszych programów, którymi będziemy się tu zajmować.

Programidło DTPiczne.

Wiele jest przyjemnych programów, które pozwalają na uprawianie DTP na Amidze. My wybierzmy najprostszy z możliwych, czyli PageSetter II. Na nim składane były pierwsze numery AMIGOWCA (również te z kolorową okładką) i dopiero potem przesiedliśmy się na bardziej profesjonalne oprogramowanie. Muszę Was uspokoić, że wiedza nabyta przy korzystaniu z tego programu uległa tylko wzbogaceniu, nigdy zaś nie straciła na wartości.

Program nie tylko, że mieści się na dwóch dyskietkach, ale całkiem znośnie można go spakować na jednym dysku, na którym zmieści się jeszcze jakiś edytor (np. CED) i kilka programów pomocniczych. Drugi dysk powinien zawierać odpowiednie czcionki.

Żeby nie było wątpliwości - czcionki do tego programu są albo typu CG (CompuGraphic) albo normalne, czyli bitowe. Odpowiednich czcionek CG wykorzystywanych przez program PageSetter nie udało się mimo wszystko spolszczyć - zanim stało się to technicznie możliwe wygasło zainteresowanie programem. A szkoda. Jeśli ktoś wie jak to zrobić prosimy o kontakt na adres redakcji. Ukazała się już nowa wersja programu: III i jak tylko do nas dotrze, postaramy się sprawdzić, czy możliwe jest wykorzystanie spolszczonych czcionek CG, z których korzystamy w redakcji. Na razie w naszym opisie zadowolimy się czcionkami bitmapowymi. Najlepszym rozwiązaniem jest stworzenie czcionek samemu. Możemy wtedy wybrać ulubioną czcionkę i jakąś jej stałą wielkość. W takim układzie trzeba poszukać kogoś, kto ma spolszczone czcionki CompuGraphic. Jeśli taką czcionkę dorwiemy, to korzystając z programu Fountine (system 2.x) lub programu IntelliFont (system 3.0) bez problemu możemy wykreować niezbędną czcionkę bitmapową. Wielkość jej powinna być 4 razy większa od wielkości czcionki, którą będziemy stosować. Dzięki temu nawet na drukarce laserowej kolegi będziemy mogli uzyskać zupełnie zadowalający efekt. Do czcionki bitmapowej musi być dołączony jeszcze specjalny plik .metric. Plik ten kreujemy przy pomocy programiku Any_Font.

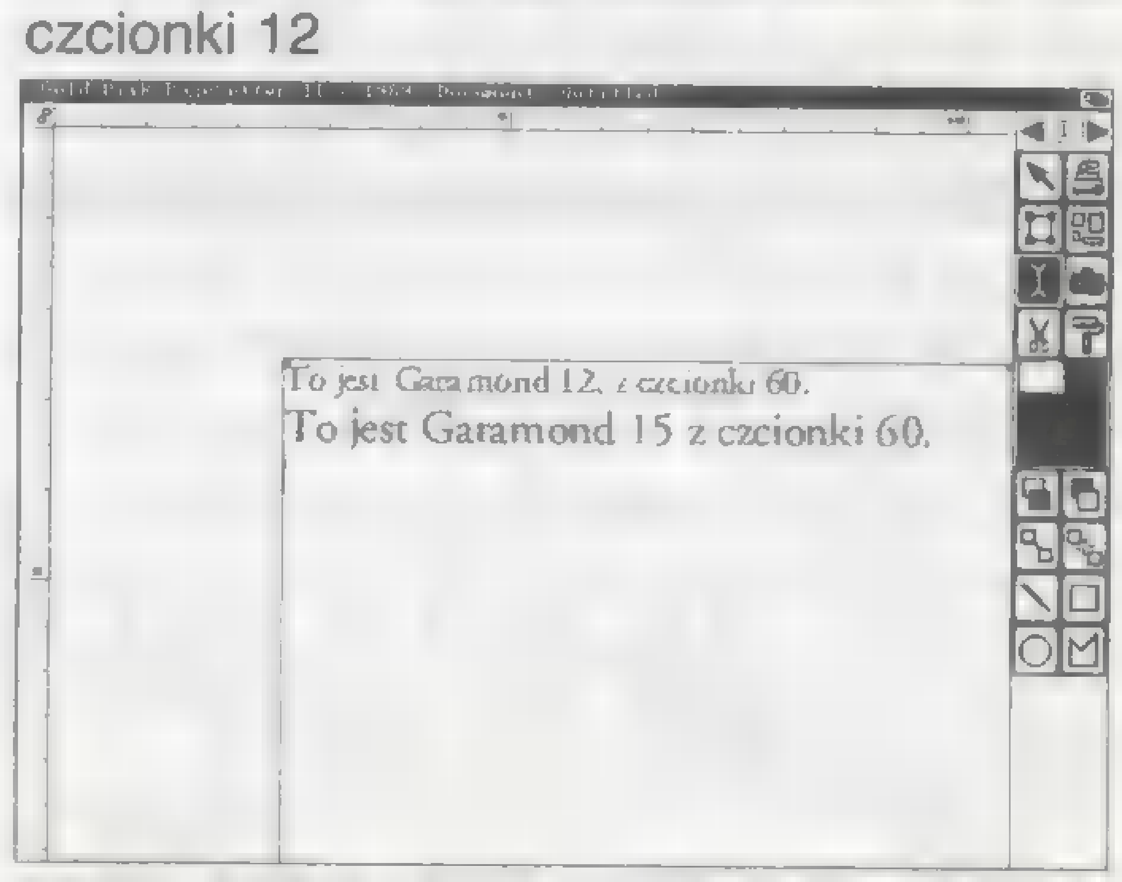
Dla przykładu: jeśli stworzyliśmy czcionkę GaramondPL o wielkości 60 punktów to powinniśmy uruchomić program Any_Font (z CLI lub Shella) w następujący sposób:

Any_Font -B 60 GaramondPL

Opcja -B 60 spowoduje, że obliczenia zostaną dokonane na właściwej wielkości czcionki i nie będziemy mieć problemów z różnego rodzaju przekłamaniami przy szerszych oknach z tekstem. Liczbę tam wpisaną należy "wypraktykować" na konkretnych przykładach.

W ten sposób uzyskaliśmy czcionkę GaramondPL do pisania wielkością 15 punktów (60/4=15). Jeśli chcemy uzyskać inne wielkości Garamondu, tak aby nie było efektów jak poniżej, to powinniśmy po prostu pozmieniać nazwy i stworzyć pliki .metric za każdym razem zmieniając wartość przy literce B np:

Any_Font -B 48 GaramondMPL, dla czcionki 12



Efekty jakich nie powinniśmy mieć.

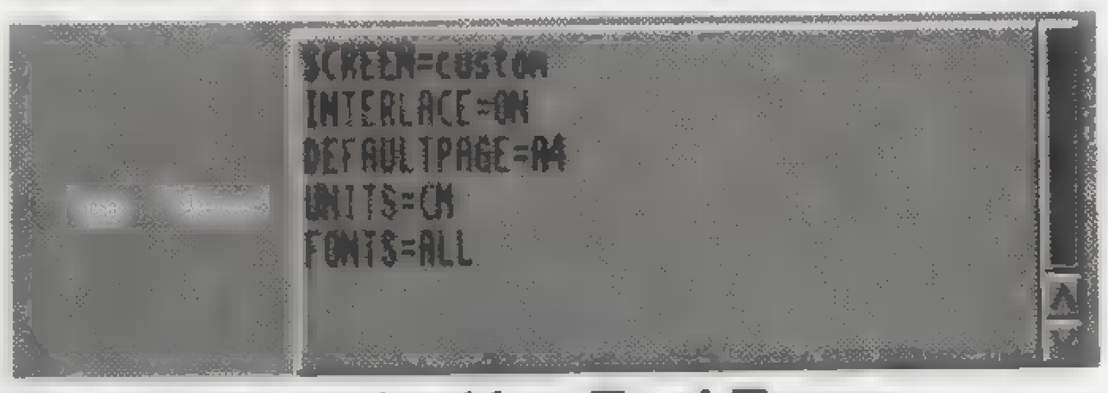
Any_Font -B 36 GaramondzikPL, dla czcionki 9, itp.

Oczywiście możemy stworzyć zestaw niezbędnych czcionek i jedynym problemem będzie miejsce na dysku. Czcionki powinny znajdować się w szufladzie fonts:, ale nie ma żadnego problemu, by przy pomocy polecenia :

Assign fonts: Czcionki:fonts

przypisać tę szufladę do dysku, który nazwiemy Czcionki i na którym w szufladzie fonts umieścimy nasze własne czcionki.

Gdy mamy czcionki możemy uruchomić program. Wystarczy kliknąć na ikonę. Dla ułatwienia dodam, że wygodnie pracuje się przy ustawionych następujących opcjach wpisanych do ikony:

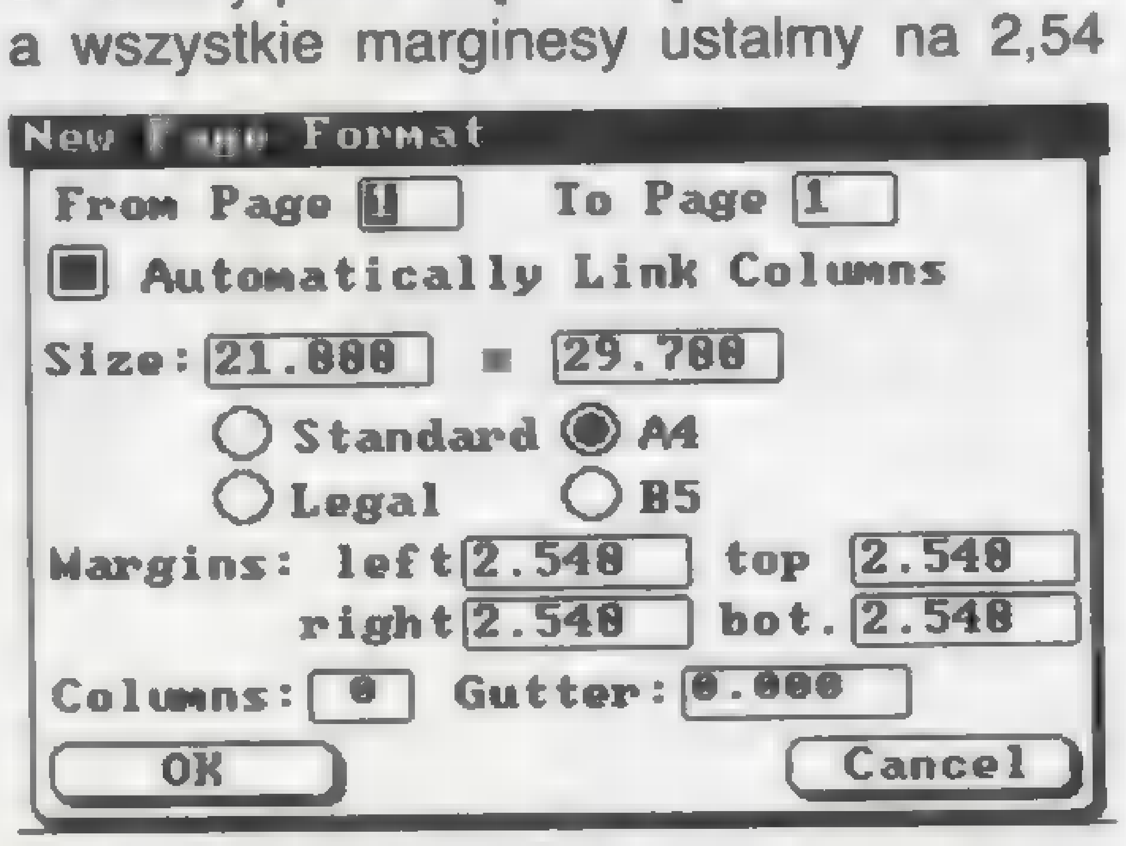


To należy zmienić w Tool Types.

Teraz uruchomi się program i będziemy mieli przed sobą czystą białą płaszczyznę.



Zanim przejdziemy do tworzenia czegokolwiek, zatrzymajmy się na kilku słowach niezbędnej teorii DTP. Jak wszyscy wiemy efektem końcowym DTP jest wydruk. Może on być wykonany po przejściu kilku stanowisk takich jak naświetlarka i drukarnia "Słowo Powszechne", może też być wykonany bezpośrednio na drukarce. O taki wydruk nam właśnie chodzi. Przeważnie drukujemy na kartkach i są one formatu A4. Oczywiście format możemy w pewnym zakresie, dowolnie wybierać - por. rysunek. Każda strona, którą drukujemy powinna być przedstawiona w naszym programie. Na początek stwórzmy pierwszą stronę w formacie A4, a wszystkie marginesy ustalmy na 2,54

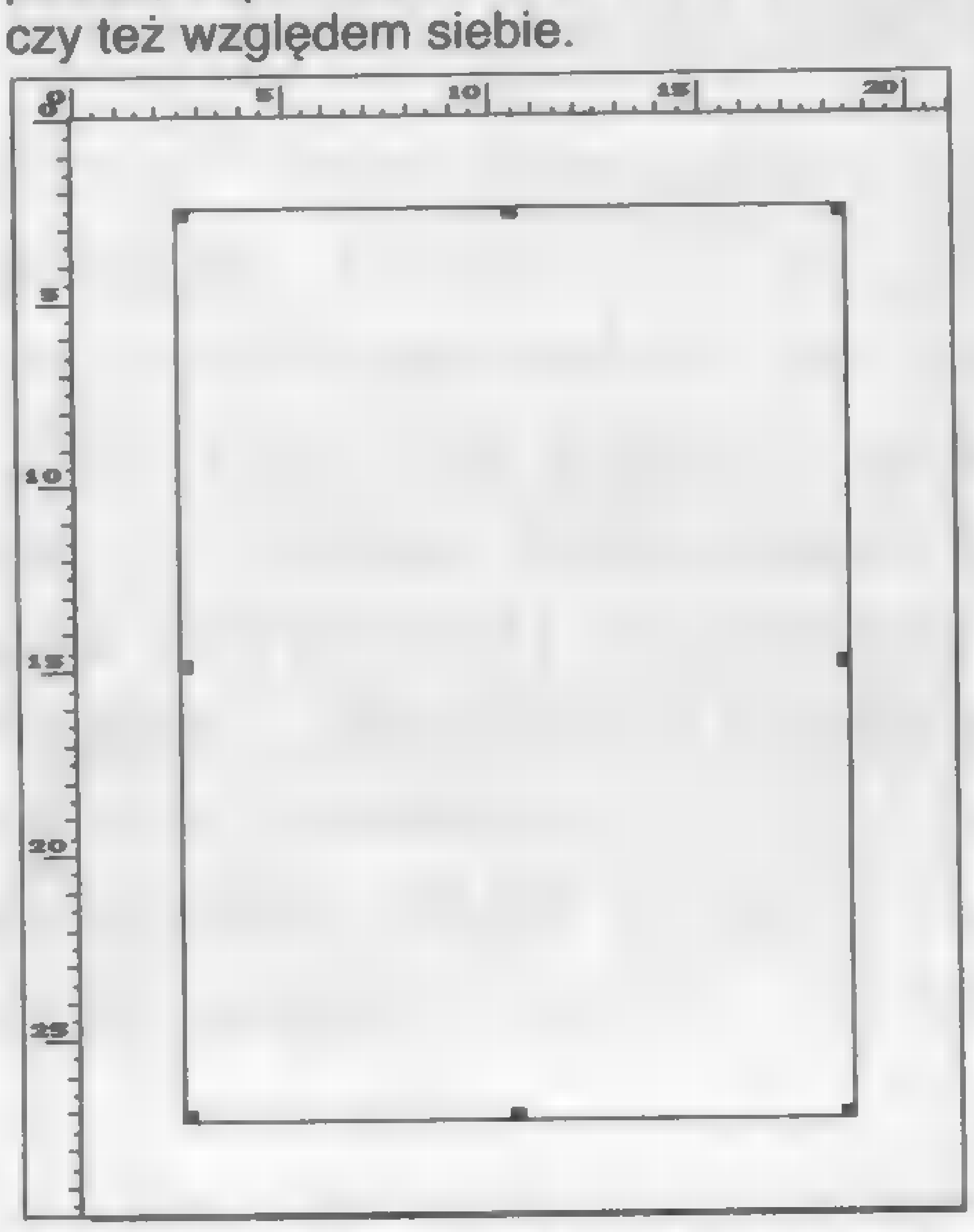


Tu wybieramy opcje.

cm (czyli 1 cal). Stronę tworzymy wybierając z menu "Page" opcję "New".

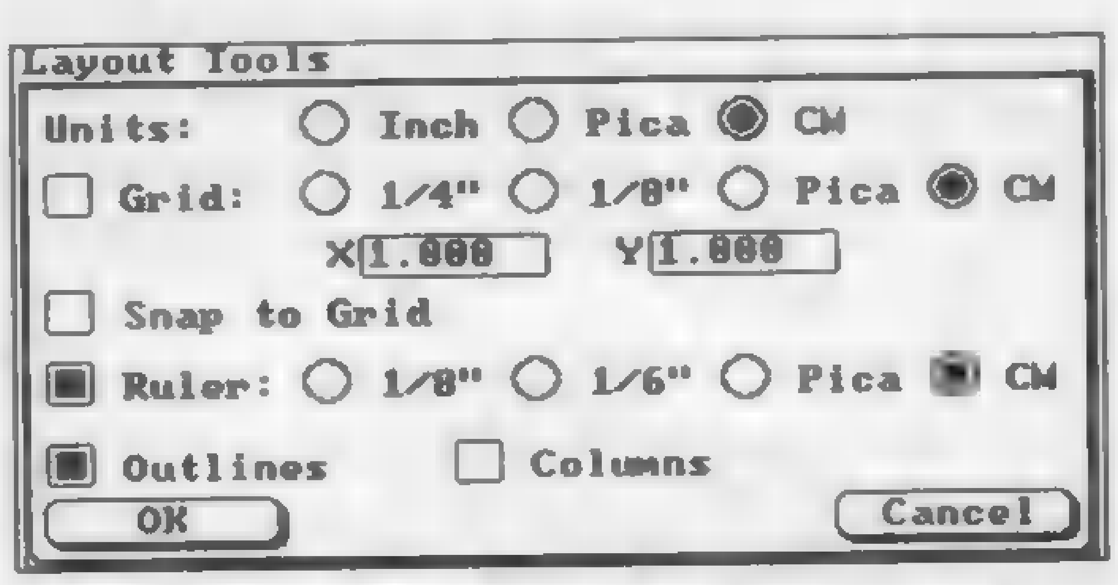
Pudełko dobre jest na wszystko

Pudełka z punktu widzenia DTP są najistotniejszymi elementami strony. W pudełkach można pisać. Do nich też można wprowadzać grafiki. Można w nich rysować. Można nimi zasłaniać niewygodne fragmenty strony itd. Na i w pudełkach dzieją się wszystkie akcje jakie przeprowadzamy. Pudełka można oczywiście grupować, odpowiednio ustawiać na stronie, czy też względem siebie.

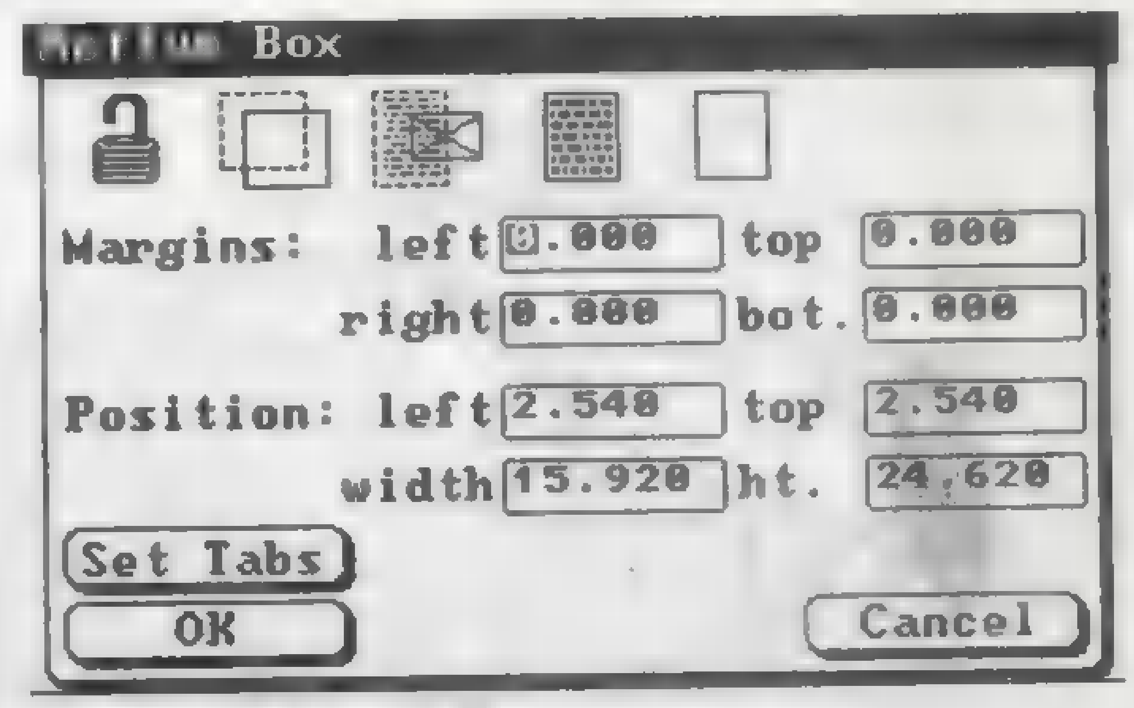


A tak wygląda nasza nowa strona wraz z pierwszym pudełkiem.

Na stworzonej stronie mamy wyświetlony ciągłą linią obrys pierwszego podstawowego pudełka. Linią przerywaną są oznaczane pudełka nieaktywne. Granice pudełka będą wyświetlane, jeżeli w menu Preferences/Layout Tools włączymy opcję Outlines.

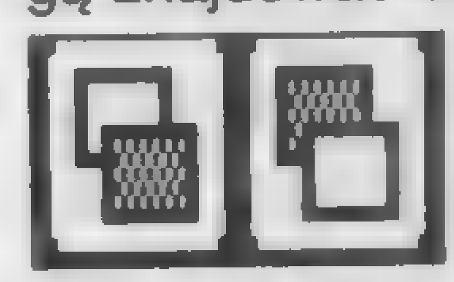


Pudełko to jest o 2,54 cm mniejsze z każdej strony niż kartka A4, ponieważ takie marginesy zadaliśmy. Do tego pudełka możemy wpisać tekst lub wczytać jakąś grafikę. W tym samym pudełku nie może być jednocześnie grafika i tekst. Pudełko oczywiście można powiększać lub zmniejszać. Służą do tego celu kwadraciki umieszczone w rogach pudełka, jak i po jego bokach. Można tego także dokonać przy pomocy specjalnej metryczki, którą wywołujemy dwukrotnym klinięciem myszką na pudełko lub z menu Box/Alter/Active.



Metryczka pudełka.

W metryczce ustalamy co ma się dziać z pudełkiem: gdzie ma się znajdować, jaką ma mieć wielkość i jak ma wpływać na tekst, który ewentualnie znalazłby się w pudełku pod nim. Pudełka bowiem mogą znajdować się na sobie lub pod sobą.



Czasami przemieszczenie pudełka jest niezbędne dla uzyskania odpowiedniego efektu z tekstem. Przemieszczanie pudełek następuje przy pomocy gadżetów: "Pudełko W Tył", "Pudełko W Przód".

Metryczka pozwala nam również na "zamknięcie" pudełka. Klikamy po prostu na rysunek kłódki i pudełko zostaje zablokowane. Tak zamknięte pudełko nie może być przemieszczane, ani nie można zmieniać jego wielkości. Zawartość oraz położenie względem innych pudełek mogą być jednak edytowane przez cały czas.

Wczytanie tekstu lub grafiki następuje z menu Projekt/Import/Text/Bitmap_Graphic/Clip. Pudełko, do którego wczytujemy grafikę musi być puste. Natomiast tekst możemy doczytywać do pudełka wypełnionego. W pierwszej fazie tekst zostaje wczytany do bufora, skąd wstawiamy go do pudełka po aktywizowaniu kursora opcją Paste. Wielkość i rodzaj czcionki ustalamy albo przed wczytaniem, albo po zaznaczeniu wybranego fragmentu tekstu (wcisnąć i przytrzymać LMB, ■ następnie zaznaczyć fragment).

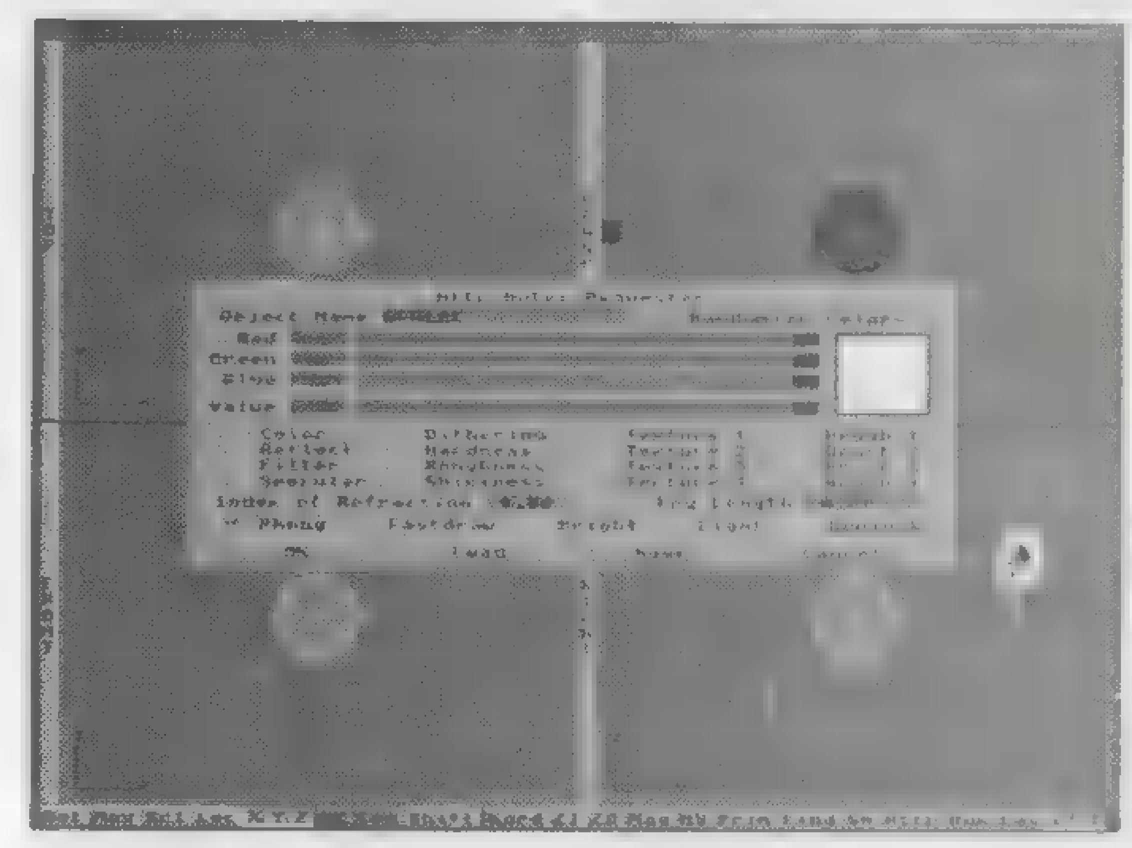
Możemy importować dwa typy grafik: bitmapowe i clipy z programu Professional Draw. Grafiki po imporcie będą się proporcjonalnie dostosowywały do wielkości pudełka. By zmienić wielkość grafiki wraz ze zmianą wielkości pudełka należy wcisnąć i przytrzymać klawisz Alt, jednocześnie zmieniając wielkość pudełka jednym ze wspomnianych wcześniej kwadracików. Ta sama operacja pozwala nam na przesuwanie grafiki wewnątrz pudełka. W programie PageSetter II jest pewna wada, która powoduje, że grafiki są czasami wyświetlane w negatywie, natomiast na wydruku wychodzą prawidłowo. Zależy to od ustalenia palety barw w grafice. Wadę można spróbować ominąć zgrywając grafikę na białym kolorze tła.

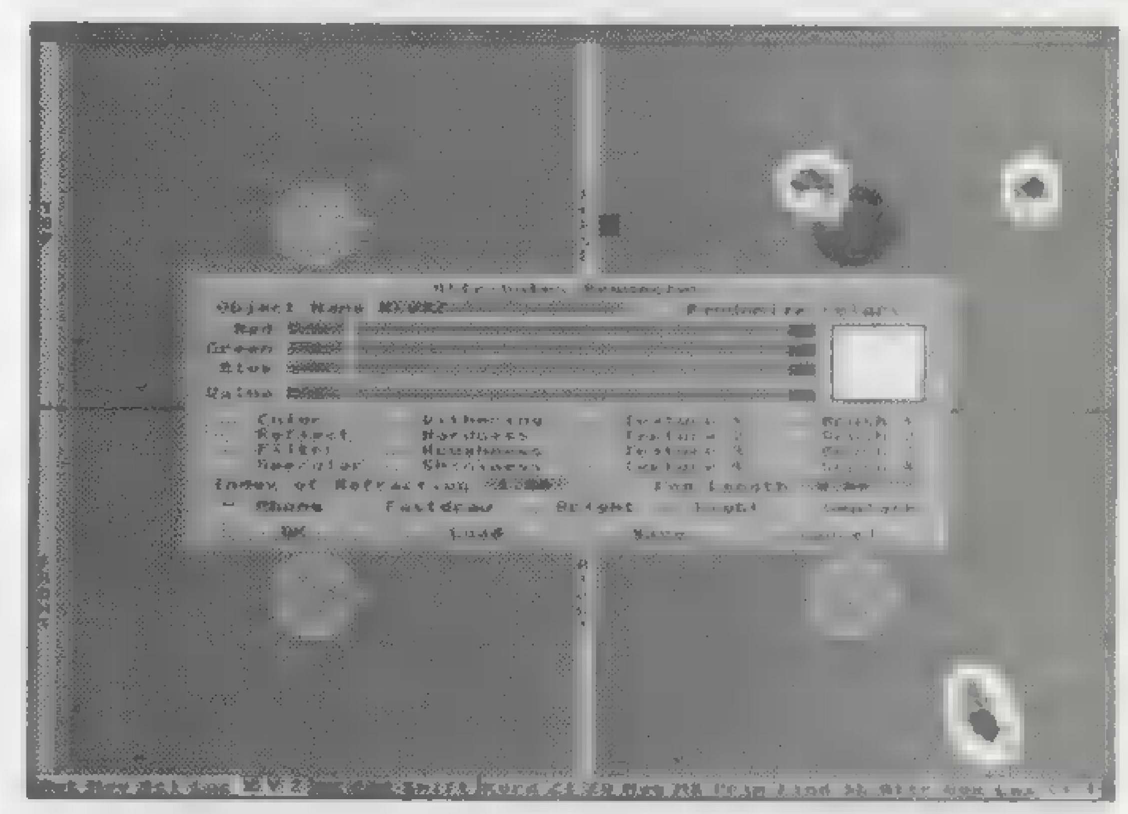
W następnym odcinku naszego kursu dla początkujących DTPistów zajmiemy się tworzeniem przykładowej strony. ❑

IMAGINE
Kurs cz.2

Marek Stor

W tym numerze Amigowca chciałbym udzielić Wam kilka rad pomocnych przy tworzeniu własnych animacji. Mam nadzieję, że już za miesiąc rozpocznę kurs o tym, jak stworzyć własną animację na Imagine. Będzie on obejmował wykonanie własnego obiektu z wykorzystaniem atrybutów oraz tekstur, wykonanie animacji w Stage i Action Editor, a także zastosowanie Cycle Editora.

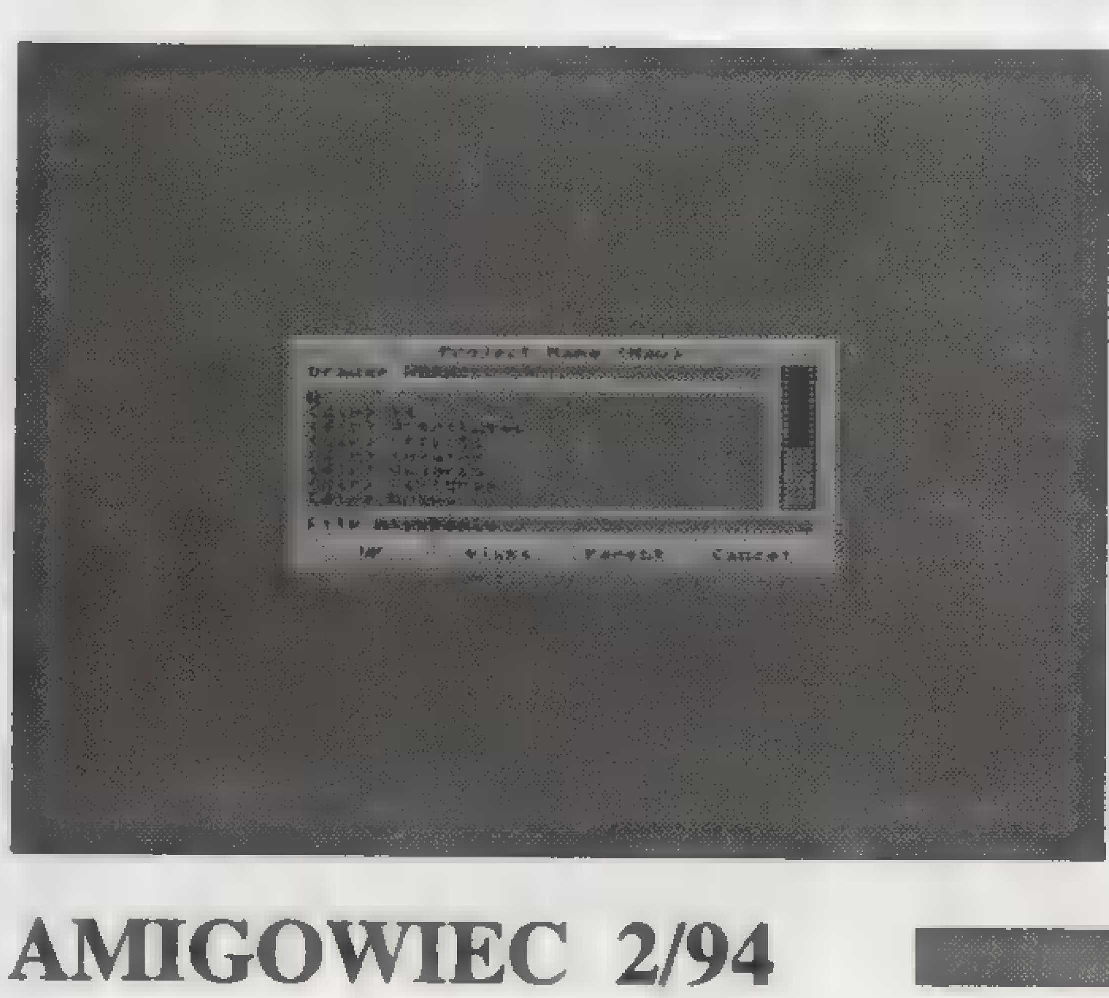




Oczywiście w trakcie kursu nadal będę przedstawiał Tobie sztuki i sztuczki. W planach jest (jeżeli naczelny Amigowca udostępni mi tyle miejsca) dział pomocy i porad dla wszystkich, którzy mają problemy nie tylko z Imagine, ale i z innymi programami animacyjnymi 3D. Na Wasze listy będą odpowiadali także moi znajomi, którzy z grafiką komputerową mają do czynienia od kilku ładnych lat. Ale już dosyć pisania, bierzemy się do roboty.

Dzisiaj chciałbym udzielić Wam kilku porad dotyczących, że tak powiem, spraw ogólnych. Będą one przydatne w trakcie kursu oraz pomogą Wam podczas tworzenia własnych animacji. Przynajmniej mam taką nadzieję.

Kiedy otwierasz nowy projekt w Imagine, nazwij go tak, jak ma się nazywać cała animacja lub scena przez Ciebie stworzona. Przyjmijmy, że chcesz przenieść do Imagine całe swoje mieszkanie. Otwierasz w PROJECT EDITOR



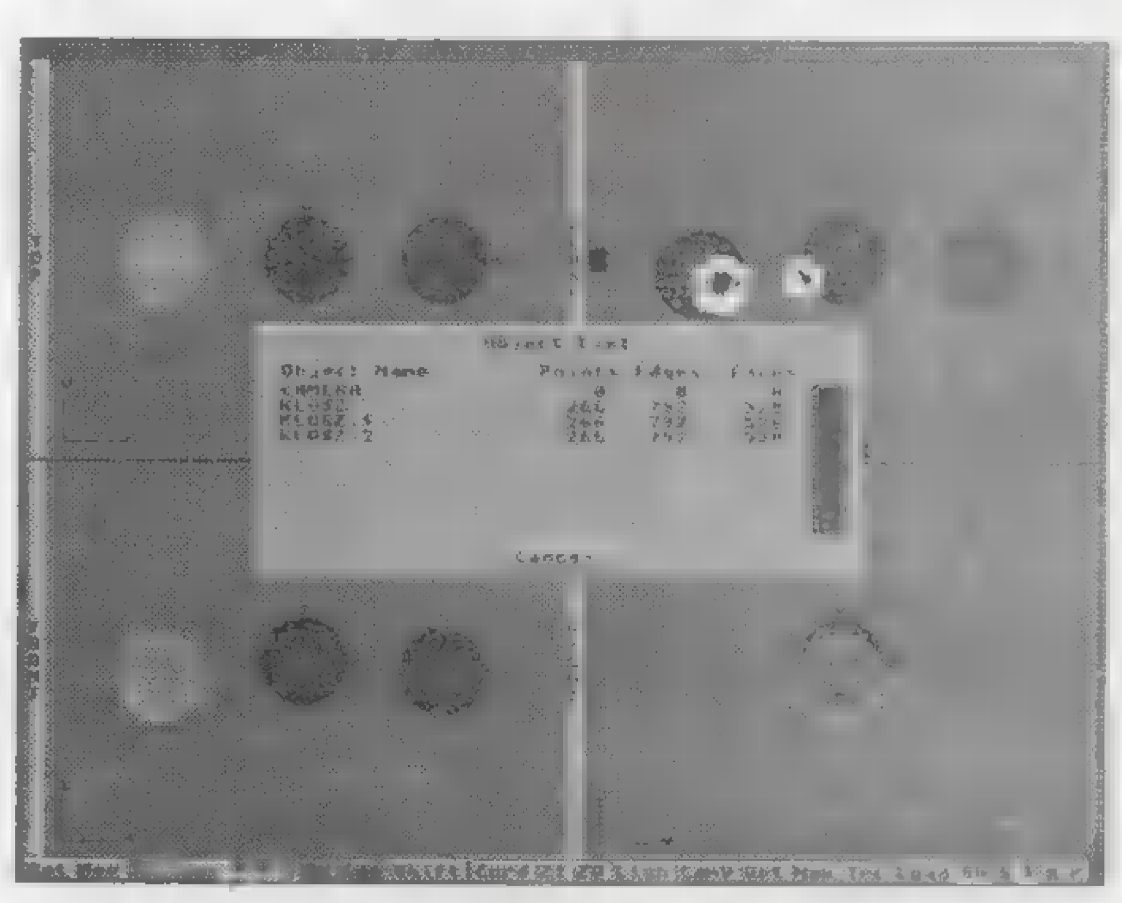
- PROJECT NEW i nadajesz mu nazwę MIESZKANIE. Imagine sam tworzy z niego szufladę o nazwie MIESZKANIE.IMP. W tej szufladzie znajduje się między innymi podkatalog OBJECTS, w którym będziesz zapisywał wszystkie stworzone obiekty.

Jeżeli tworzysz jakiś obiekt w swoim projekcie za pomocą DETAIL EDITORA, musisz zwrócić szczególną uwagę na nazwę obiektu. Budowa każdego obiektu, który tworzysz bez wykorzystania gotowych brył, zaczyna się od ustawienia jego osi (AXIS). Imagine sam przypisuje nazwę tworzonego obiektu. I tak wszystkie obiekty stworzone od początku nazywają się AXIS. Identycznie obiekty tworzone przy pomocy gotowych brył przyjmują nazwy tych brył np. wszystkie obiekty stworzone z kuli będą miały nazwę SPHERE. Już to widzę jak zastanawiasz się przy montowaniu animacji czy AXIS.1 to jest stół w kuchni, czy twoje biurko.

W DETAIL EDITOR nazwę obiektu zmieniamy w funkcji ATRIBUTTES.

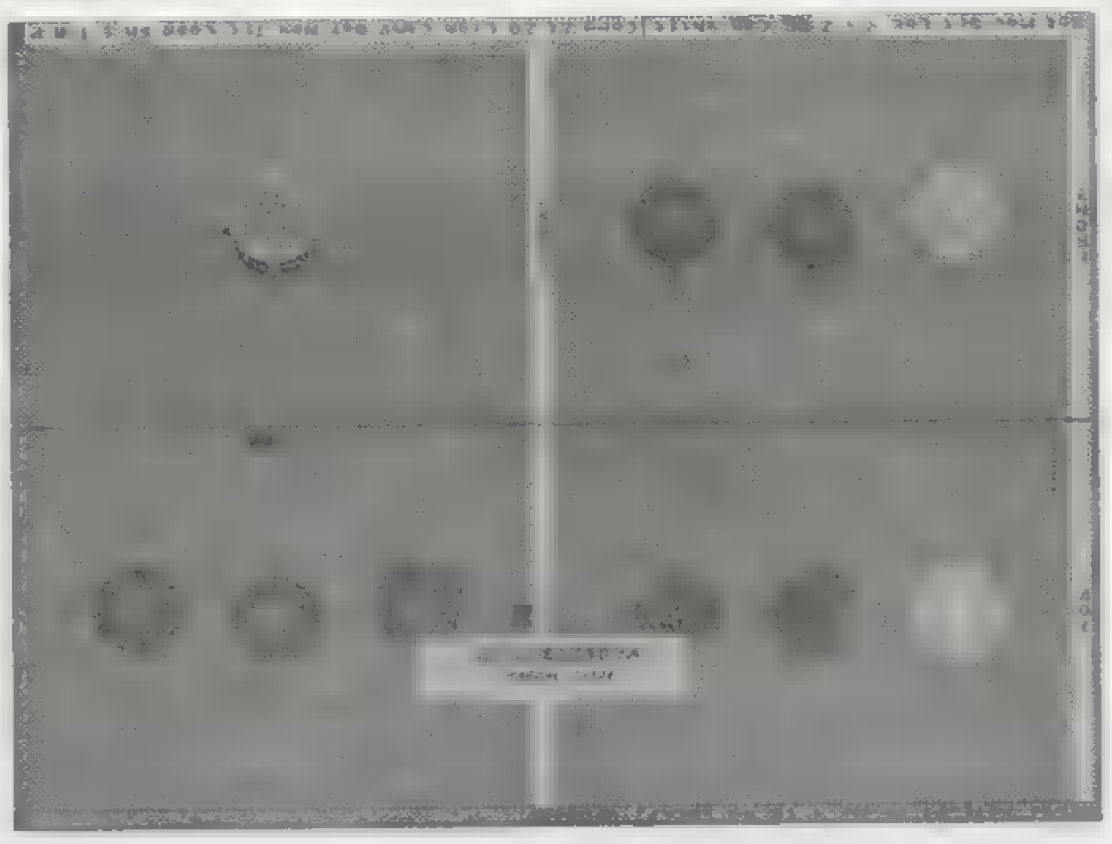
Dla przykładu chcesz zmienić nazwę obiektu stworzonego z kuli, a będącego w naszej animacji kloszem na lampę. Nie wystarczy nadać mu nazwy "klosz" podczas nagrywania go na dysk. W Action Editor jego nazwa pozostanie nie zmieniona. Trzeba otworzyć funkcję ATRIBUTTES i w oknie OBJECT NAME zmienić nazwę SPHERE na KLOSZ.

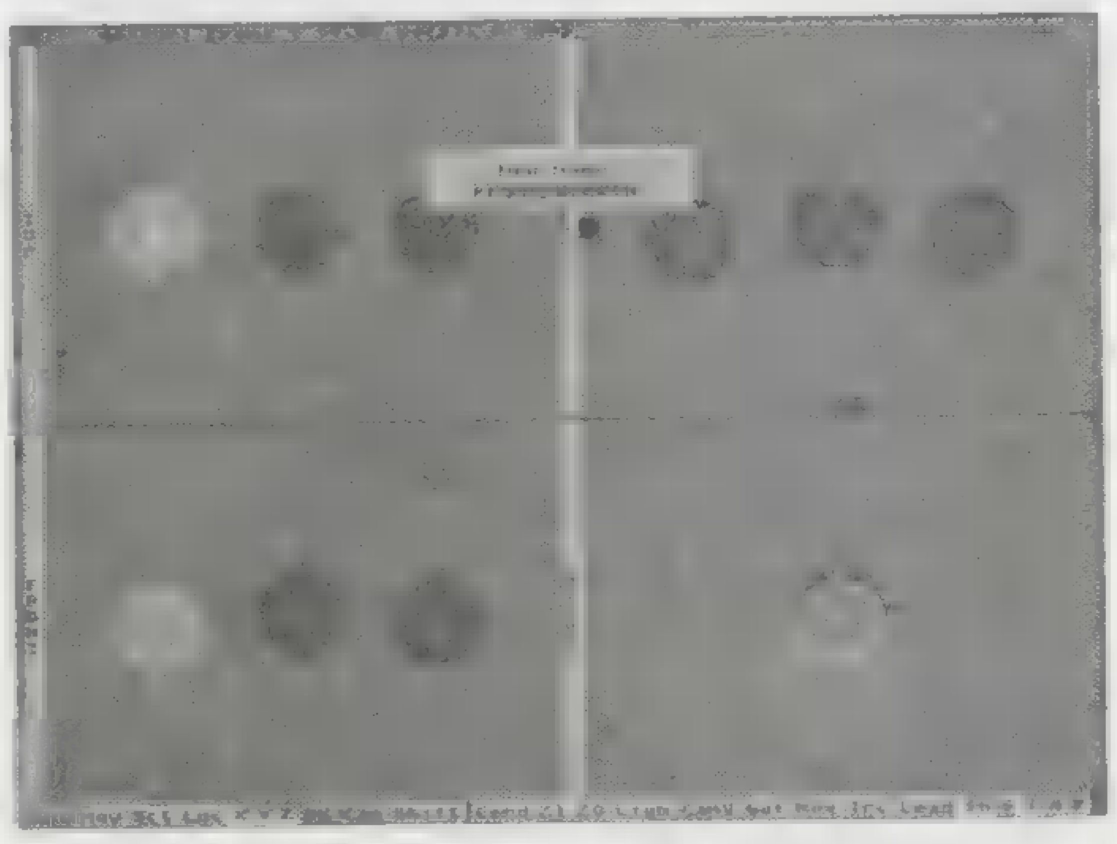
Po raz drugi możesz zmienić nazwę obiektu w STAGE EDITOR. Najlepiej zobrazuje to przykład. Wgraj w STAGE EDITOR trzy razy ten sam obiekt "klosz". Uzyskasz trzy obiekty o takich samych nazwach i ponumerowane: KLOSZ, KLOSZ.1, KLOSZ.2. Przypuszczam, że nie wiesz, który z tych kloszy jest do kuchni, który do przedpokoju, a który do łazienki.

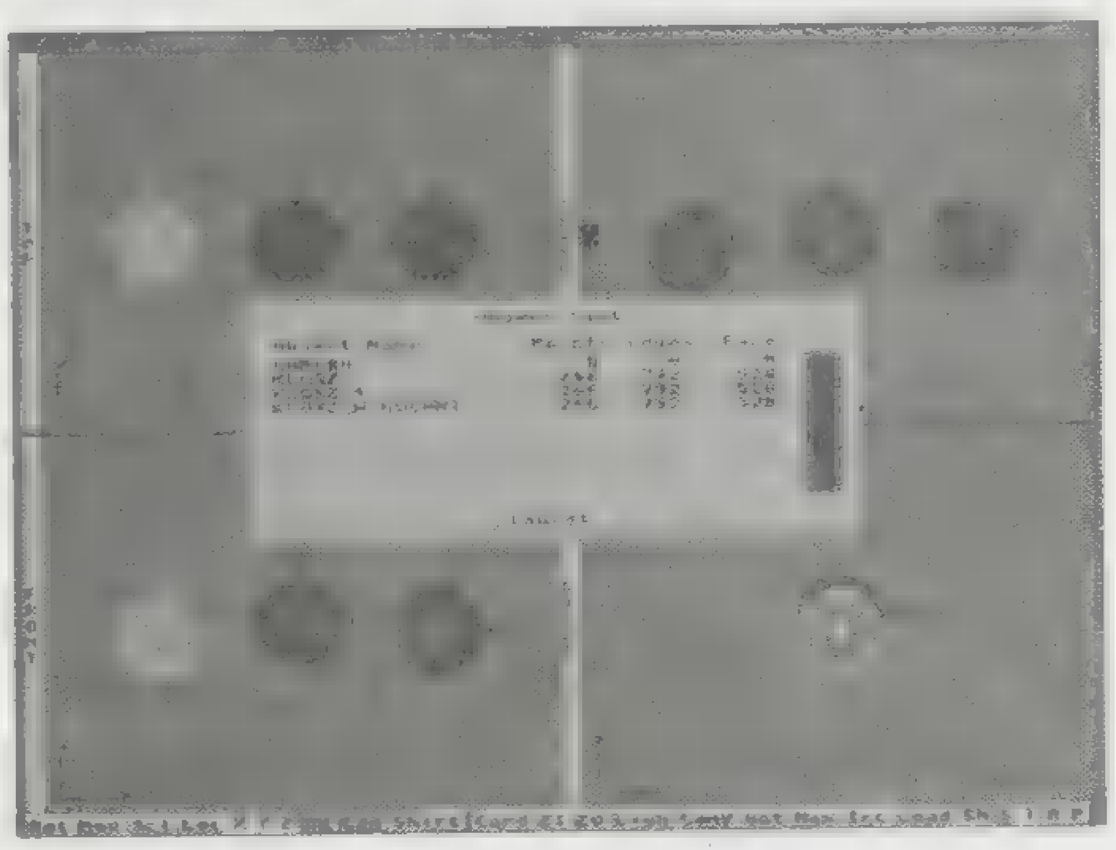


Dochodzenie czy KLOSZ.2 jest do kuchni, czy do łazienki jest łatwe przy

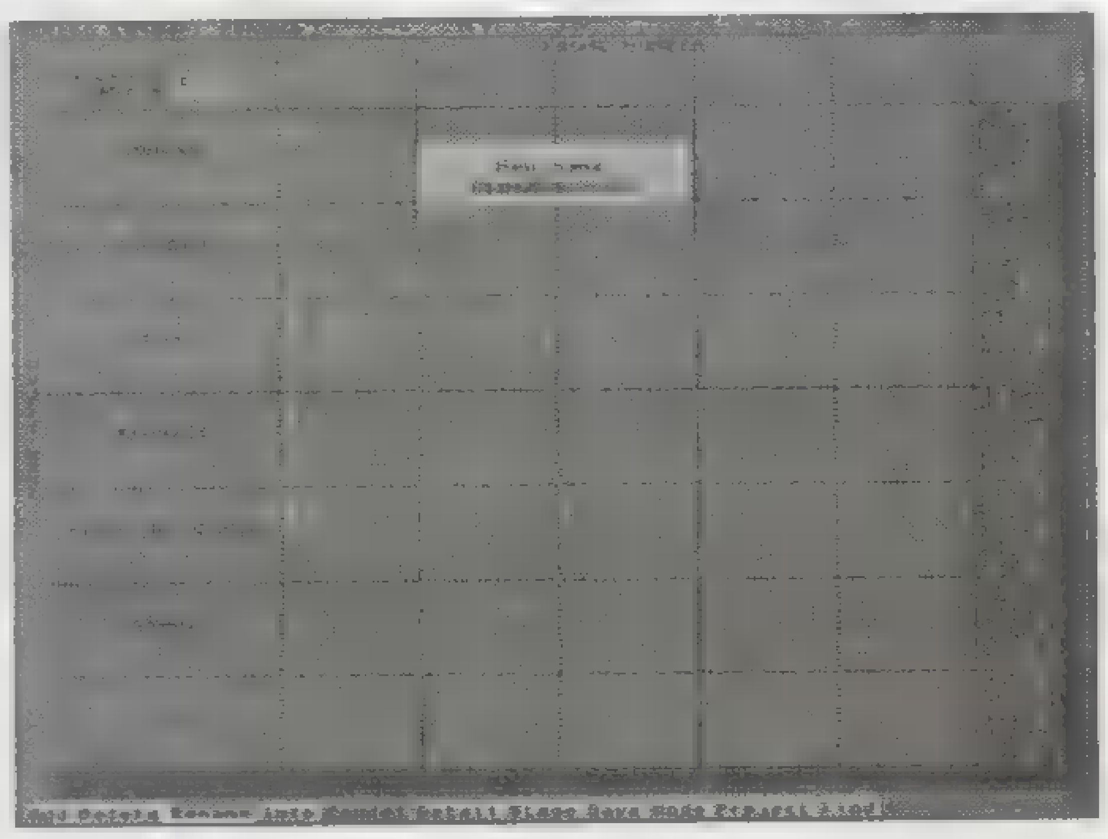
trzech obiektach. A co w momencie, gdy będziesz miał tych obiektów 20. Aby zmienić nazwę objektu otwórz z górnego menu funkcję OBJECT/RENAME. Pokaże się requester NEW NAME z wpisaną nazwą KLOSZ.2. Zmień ją na KLOSZ_W_KUCHNI. Teraz już nie powinieneś mieć problemu ze znalezieniem odpowiedniego obiektu w gotowej scenie.

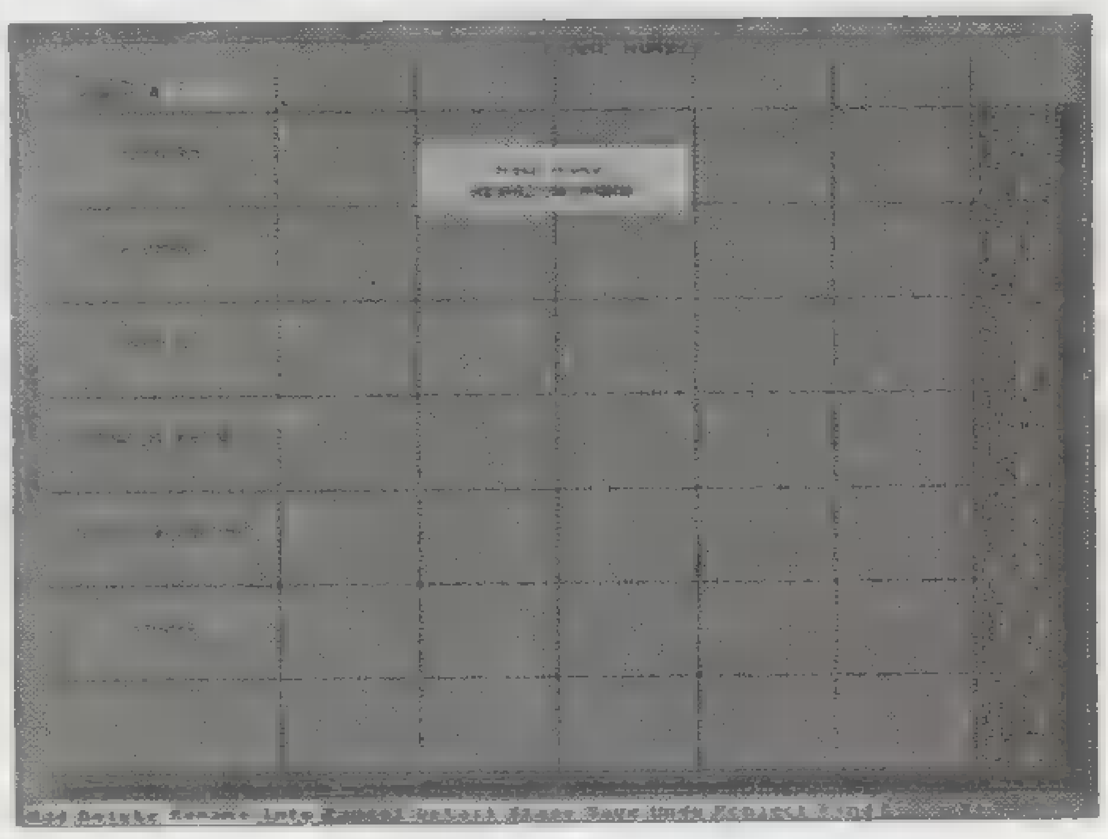




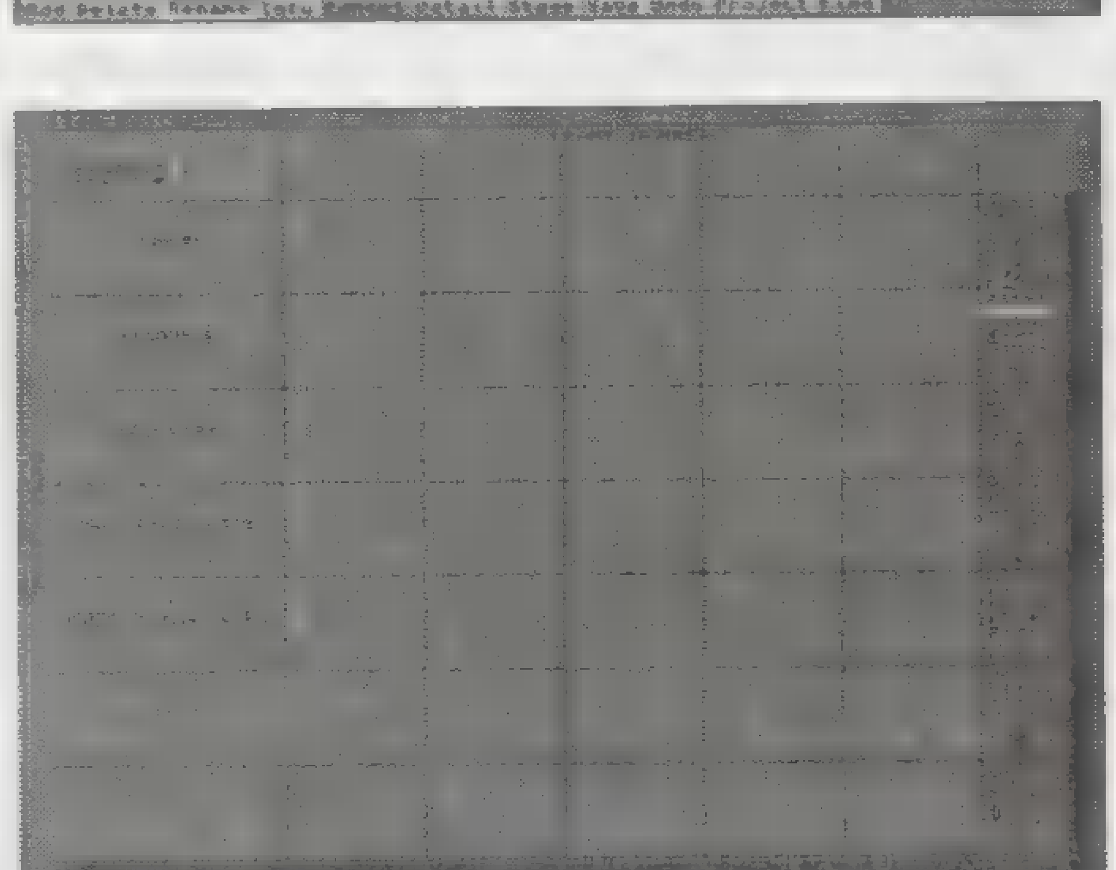


Możesz jeszcze raz dokonać zmiany nazwy obiektu, a także uporządkować obiekty według nazwy. Ma to zastosowanie przy bardzo rozbudowanych animacjach, gdzie do jednego obiektu przypisanych jest kilka innych rzeczy. Nazwę obiektu zmienia się wywołując z górnego menu FUNKCTIONS/RENAME, a sortuje przy użyciu funckji FUNKCTIONS/SORT. I tak dla przykładu, nasz główny obiekt nazywa się MOTOCYKL. Jeżeli nasz motocykl ma poruszać się po scieżce, to niech scieżka nazywa się MOTOCYKL.PATH. Jeżeli chcesz do naszego motocykla dodać kolejne scieżki np. kiwanie na boki podczas brania zakrętu, niech scieżka ta nazywa się MOTOCYKL.ZAKRĘT. Funkcja SORT porządkuje wszystkie obiekty i scieżki rozpoczynające się od słowa motocykl w kolejności alfabetycznej.









PAMIĘTAJ! Staraj się, aby wszystkie obiekty do danego projektu były w jednym podkatalogu. Możesz w nim stworzyć dowolną ilość podkatalogów np. dla projektu MIESZKANIE.IMP w podkatalogu OBJECTS stwórz podkatalogi POKÓJ MAŁY, POKÓJ DUŻY, KUCHNIA, itp. Ułatwi Ci to szukanie danego obiektu. Równocześnie, gdy będziesz chciał zachować projekt w swoich archiwach, możesz być spokojny, że niczego nie pominąłeś. To samo dotyczy brushy, ścieżek i innych rzeczy wykorzystanych w projekcie. Staraj się, aby nazwy obiektów były jak najkrótsze. Jeżeli zastosujesz powyższe zasady bądź pewny, że w szufladzie "MIESZKANIE.IMP" znajdują się wszystkie niezbędne elementy do wygenerowania animacji.

Druga porada dotyczy sposobu przyspieszenia podglądu animacji w STAGE EDITOR. Na ideę tego pomysłu natrafiłem przyglądając dysk Public Domain do Imagine.

Obiekty wykorzystywane w naszych animacjach są coraz bardziej rozbudowane i zajmują coraz więcej "kilo" np. objekt statku U.S.S. ENTERPRISSE z filmu Star Trek ma 1.5 MB. Spróbuj w STAGE EDITORZE dopasować ścieżkę, po której ma poruszać się obiekt. Twoja animacja składa się ze 125 klatek i chcąc mieć pewność, że ruch statku będzie płynny dajesz w menu ANIMATE/MAKE, by program obliczał każdą klatkę. I co? Nie dość, że na efekt końcowy czekasz sporo czasu (ładowanie obiektu o takiej wielkości nawet z dysku twardego trochę trwa), to jeszcze niepotrzebnie "męczysz" swój dysk. Jest na to prosta metoda.

1. Wgraj w DETAIL EDITOR obiekt statku kosmicznego.
2. Wywołaj z FUNKCTIONS/ADD/PRIMITIVE lub naciśnij F5. Wybierz walec lub płaszczyznę. Jeżeli obiekt ma mieć kształt sześcianu wybierz PLANE (płaszczyznę) i wytłocz ją funkcją EXTRUDE.
3. Następnie używając opcji skalowania dopasuj sześcian lub walec do naszego obiektu, tak by miały identyczną wielkość. UWAGA!!! Osie obiektu muszą się zgadzać ze sobą.
4. Nagraj nowy obiekt na dysk pod nazwą np. Statek.proba.
5. Wstaw go w miejsce statku w STAGE EDITOR.
6. Teraz możesz dowolnie manipulować ścieżką, a czas podglądu animacji skróci się kilkakrotnie.
7. Gdy twoja ścieżka będzie już gotowa, podmień tylko obiekty w ACTION EDITOR i możesz już generować całą animację.

Uff. Myślę, że macie co robić przez następny miesiąc, a ja z niecierpliwością czekam na Wasze listy. Jestem ciekawy, czy to co piszę w ogóle kogoś interesuje? ❑

SUPRA FAX-MODEM 14400

Tomasz Matajek

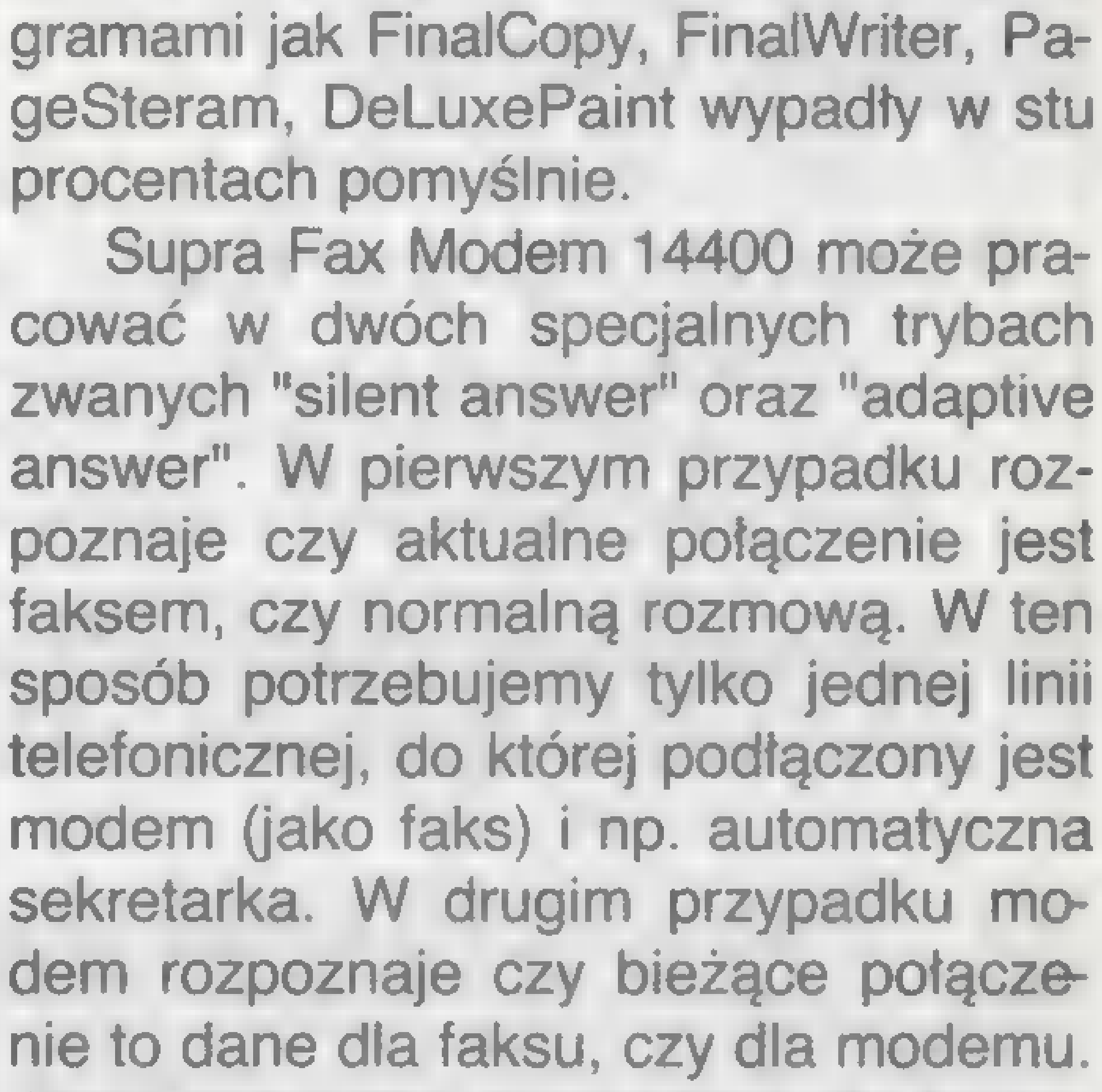
Firma SUPRA jest dobrze znana na rynku Amigi już od kilku lat. Produkuje rozszerzenia pamięci, kontrolery twardych dysków, karty turbo, no i oczywiście modemy. Standardowym modemem tej firmy jest model v32bis. V32bis jest to po prostu nazwa międzynarodowego standardu CCITT definiującego połączenia modemowe przy prędkości 14400 (częstotliwości nośne, rodzaj i głębokość modulacji, itd).

W tym miejscu warto zwrócić uwagę na fakt, iż "modem modemowi nie równy". Na rynku jest dostępnych kilka odmian tego urządzenia różniących się numerem wersji wewnętrznego systemu operacyjnego zawartego w pamięci ROM. Wersje z ROMem 1.7xx i 1.8xx działają bez porównania lepiej niż starsze modele. Różnice dotyczą przede wszystkim stabilności połączeń oraz rozpoznawania sygnału zajętości. Użytkownicy modemów ze "starą" wersją ROMu (poniżej 1.7xx) nie mają się jednak czym martwić, ponieważ wymiana ROMu na nowy jest sprawą bardzo prostą. Można to załatwić bezpośrednio z producentem za około 30 dolarów USA (jest to jedyny legalny sposób) lub przez samodzielne zaprogramowanie odpowiednich EPROMów. Należy jednak pamiętać, że do najstarszych modeli pasuje ROM 1.8xx, a do nowszych 1.7xx, gdyż ich płytki nieco się różnią. Obie wersje są jednak funkcjonalnie całkowicie równoważne.

Obudowa modemu jest aluminiowa, z estetycznie wykonaną czarną ścianką przednią, na której znajduje się łatwo dostępny wyłącznik, cztery sygnalizacyjne diody LED oraz wyświetlacz alfanumeryczny - rzecz raczej nie spotykana w innych modemach. Wyświetlacz ten jest bardzo użyteczny. Informuje na bieżąco o stanie w jakim znajduje się modem, np. pokazuje aktualną prędkość połączenia, rodzaj użytej kompresji i korekcji błędów, a także inne informacje. Modem automatycznie wybiera najlepszy osiągalny w danej chwili rodzaj połączenia w zależności od stanu linii telefonicznej i możliwości drugiego modemu. W przeciwieństwie do poprzednich wersji tego urządzenia, uzyskane połączenie jest stabilne i praktycznie bardzo rzadko zdarza się jego "zerwanie". Zależy to oczywiście od jakości modemu po drugiej stronie linii. Jeśli w trakcie transmisji stan linii ulegnie pogorszeniu, modem przełącza się automatycznie na niższą prędkość. Po ustaniu zakłóceń ponownie przestawi się na pracę z maksymalnie dostępną w danej chwili szybkością. Modem "rozumie" wszystkie określone normą CCITT rodzaje kompresji (MNP5, LAPM) i korekcji błędów (MNP2, 3, 4, 10). Aby możliwe było pełne wykorzystanie zalet kompresji, połączenie komputer-modem zaleca się ustawić na 57600. Dostarczana z modemem instrukcja w języku angielskim jest tak przejrzyście napisana, że po jej przeczytaniu każdy bez specjalnych problemów będzie umiał ustawić modem stosownie do indywidualnych potrzeb.

Jak wynika z nazwy modem może pracować jako telefaks i to z prędkością 14400. Niezbędne jest do tego celu odpowiednie oprogramowanie. Modem można nabyć w dwóch wesjach - w stanie "surowym" oraz "modem + oprogramowanie" specjalne do Amigi. W drugim przypadku modem kosztuje około 40 dolarów więcej, ale w opakowaniu znajduje się komplet kabli, a ponadto stajemy się w pełni legalnym użytkownikiem znakomitego programu GPFax (oczywiście wraz z instrukcją). Pisanie faxów jest tak proste jak to tylko możliwe. Dokument można stworzyć w dowolnym programie, który przy drukowaniu korzysta z driverów Workbencha. GPFax automatycznie przechwytuje dane idące na drukarkę i zamienia je na fax. Próby przeprowadzone z takimi programami jak FinalCopy, FinalWriter, PageSteram, DeLuxePaint wypadły w stu procentach pomyślnie.

Supra Fax Modem 14400 może pracować w dwóch specjalnych trybach zwanych "silent answer" oraz "adaptive answer". W pierwszym przypadku rozpoznaje czy aktualne połączenie jest faksem, czy normalną rozmową. W ten sposób potrzebujemy tylko jednej linii telefonicznej, do której podłączony jest modem (jako faks) i np. automatyczna sekretarka. W drugim przypadku modem rozpoznaje czy bieżące połączenie to dane dla faksu, czy dla modemu. Opcja przydatna szczególnie dla "sysopów" prowadzących własne BBSy, gdzie linię telefoniczną wykorzystuje się wyłącznie dla modemu, a mimo to nadal istnieje możliwość odbierania faksów. Opisywany modem ma jeszcze inne ciekawe cechy, jak np. CallerID, umożliwiający identyfikację numeru roz



mówcy jeszcze przed podniesieniem słuchawki. Niestety, potrzebne są do tego specjalne centrale telefoniczne niedostępne w Polsce.

Ewentualni nabywcy modemu muszą się liczyć z koniecznością zakupu właściwego zasilacza (9V AC, 650 mA), gdyż oryginalny przeznaczony jest do amerykańskiej sieci zasilającej 110V/60 Hz. Nie jest to olbrzymi wydatek - około 100.000 złotych. Cena modemu wraz z oprogramowaniem do Amigi i kompletem kabli stanowi mniej więcej równowartość 360 dolarów USA. Biorąc pod uwagę wszystkie jego zalety oraz pięcioletnią gwarancję, można uznać, iż jak najbardziej jest warty swojej ceny. ❑

SAMPLOWANIE

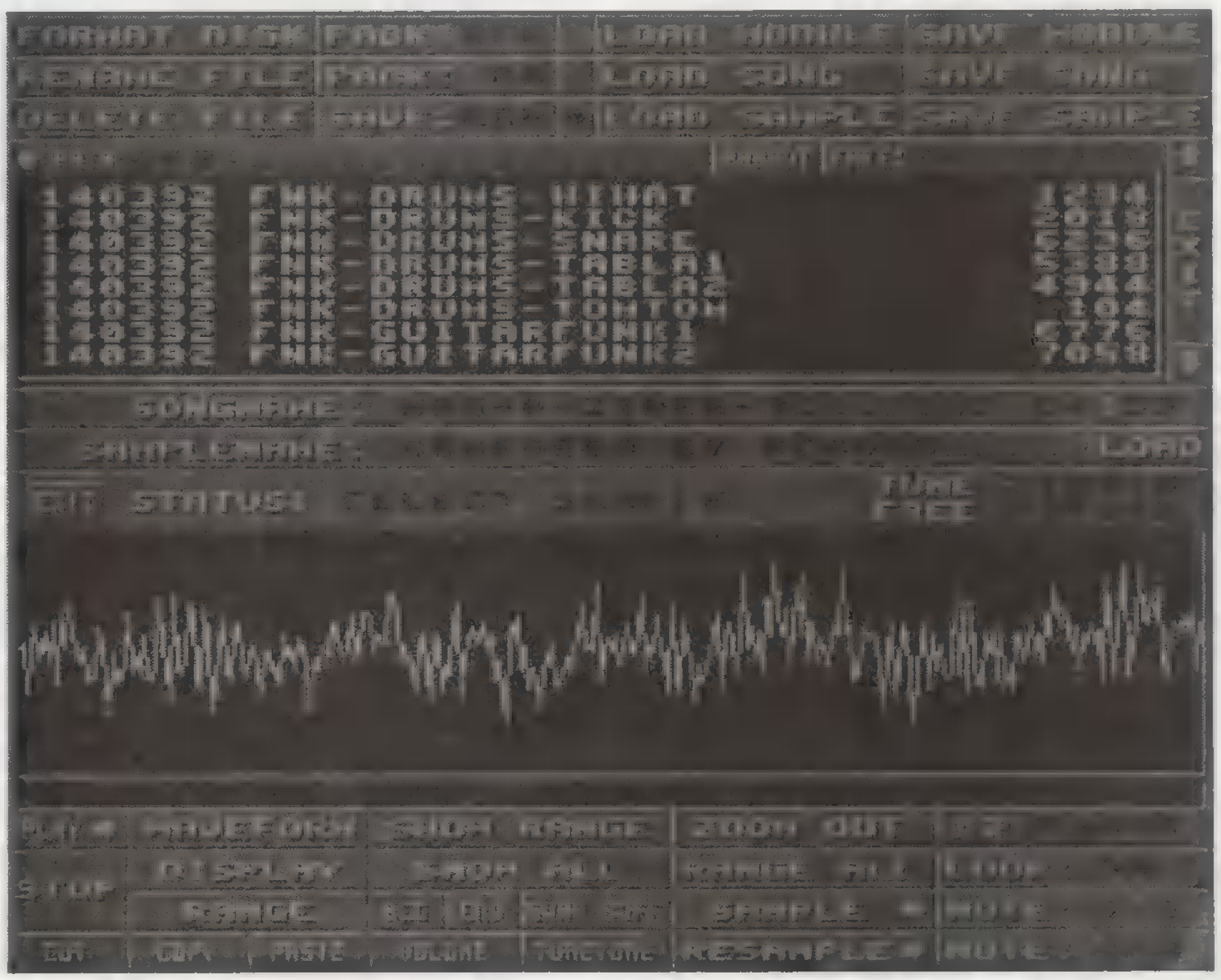
Marcin Gackowski

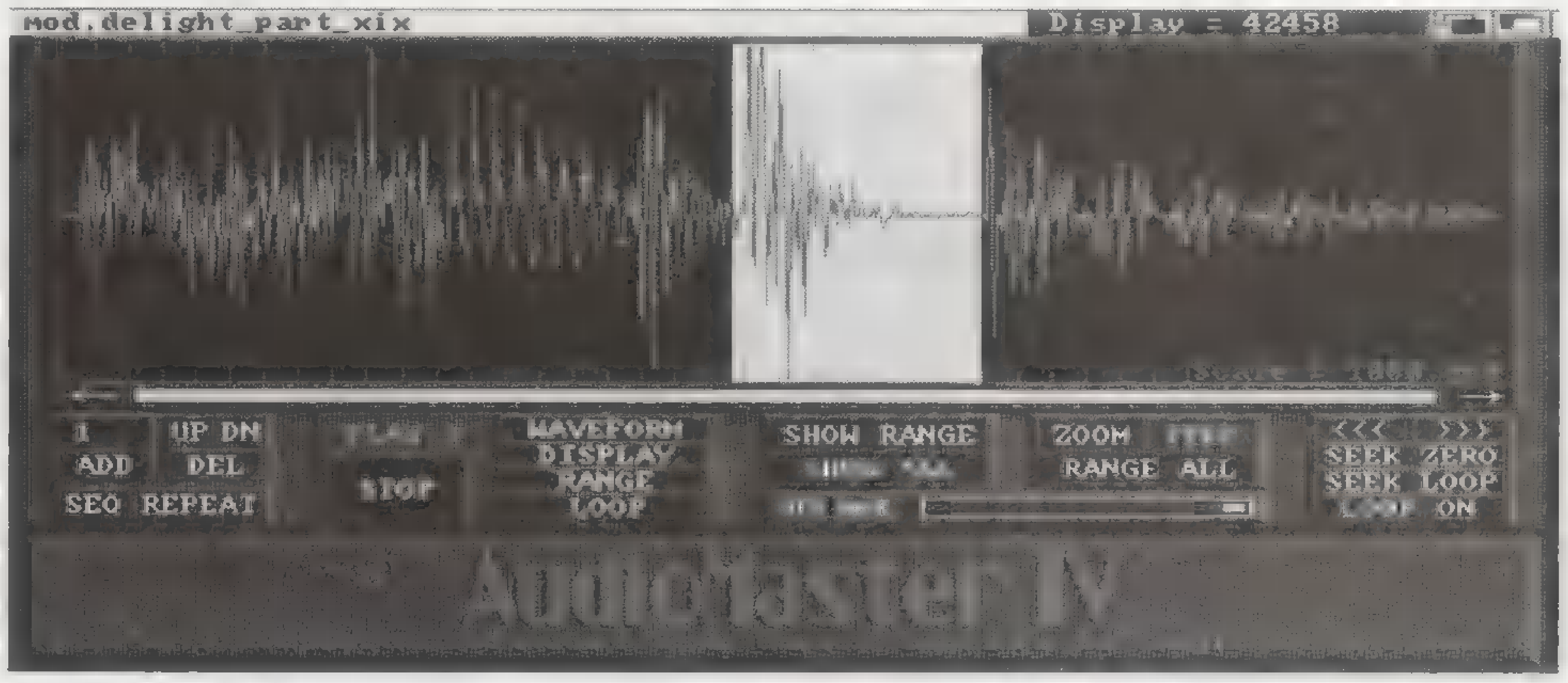
Ten odcinek kącika dla muzyków postanowiłem poświęcić samplowaniu. Jest to wbrew powszechnie panującej opinii czynność łatwa, choć nie zawsze przyjemna i dająca satysfakcjonujące nas wyniki. Warto wspomnieć, że mimo ogromnego wyboru zarówno sprzętu jak i oprogramowania, niewielu polskich muzyków sampluje samodzielnie. A jeżeli już, to tylko okazjonalnalnie.

Fakt ten najczęściej tłumaczy się brakiem instrumentu klawiszowego, ja jednak uważam, że główną przyczyną ignorowania samplera przez polskich muzyków jest lenistwo. A efektem - mnóstwo plagiatów i zwyczajnego złodziejstwa. Naprawdę niedużo potrzeba, aby to zmienić. Średniej klasy odtwarzacz CD lub przynajmniej dobry magnetofon naprawdę wystarcza do rozpoczęcia zabawy z samplowaniem...

Nie chciałbym się w tym miejscu rozwodzić nad zaletami i wadami poszczególnych modeli samplerów dostępnych na rynku, bowiem to - jaki kupimy, będzie zależało jedynie od ilości posiadanych pieniędzy. W przypadku samplerów reguła - im droższy tym lepszy - właściwie się potwierdza. Droższe samplery są zbudowane na lepszych i szybszych elementach, tańsze bazują na najekonomiczniejszych rozwiązaniach technicznych rzucających niestety cień na możliwości i parametry.

Mamy już sampler i wiele dobrych chęci. Należy teraz zastanowić się nad wyborem programu na którym będziemy tworzyć i obrabiać instrumenty. Programy takie jak: Audiomaster, Samplitude, Amas, Deluxe Sound czy Digiton oferują podobne możliwości i właściwie każdy z nich wydaje się być wystarczającym. Należy pamiętać o tym, że stworzone sample będą użyte jako instrumenty w Protrackerze, więc nie mogą być dłuższe niż 64 KB. Oczywiście sam Protracker znakomicie nadaje się do samplowania. Moduł obsługujący sampler automatycznie przerywa samplowanie po zapełnieniu ostatniego z 65535 bajtów. Stworzony instrument możemy na przykład zresamplować, zmiksować z innym, odwrócić, zmienić głośność i oktawę, a nawet stworzyć z niego akord. Edytor samplera jest nieco uboższy niż w wyżej wymienionych programach. Oferuje on jednak wycinanie, kopiowanie, powiększanie, wklejanie sampli bądź ich kawałków. Mankamentem samplera w Protrackerze jest brak niektórych efektów dostępnych w wyspecjalizowanych programach przeznaczonych wyłącznie do samplowania. Dobrym rozwiązaniem jest według mnie samplowanie za pomocą Protrackera, a gdy zachodzi potrzeba, podrasowanie sampla innym programem.





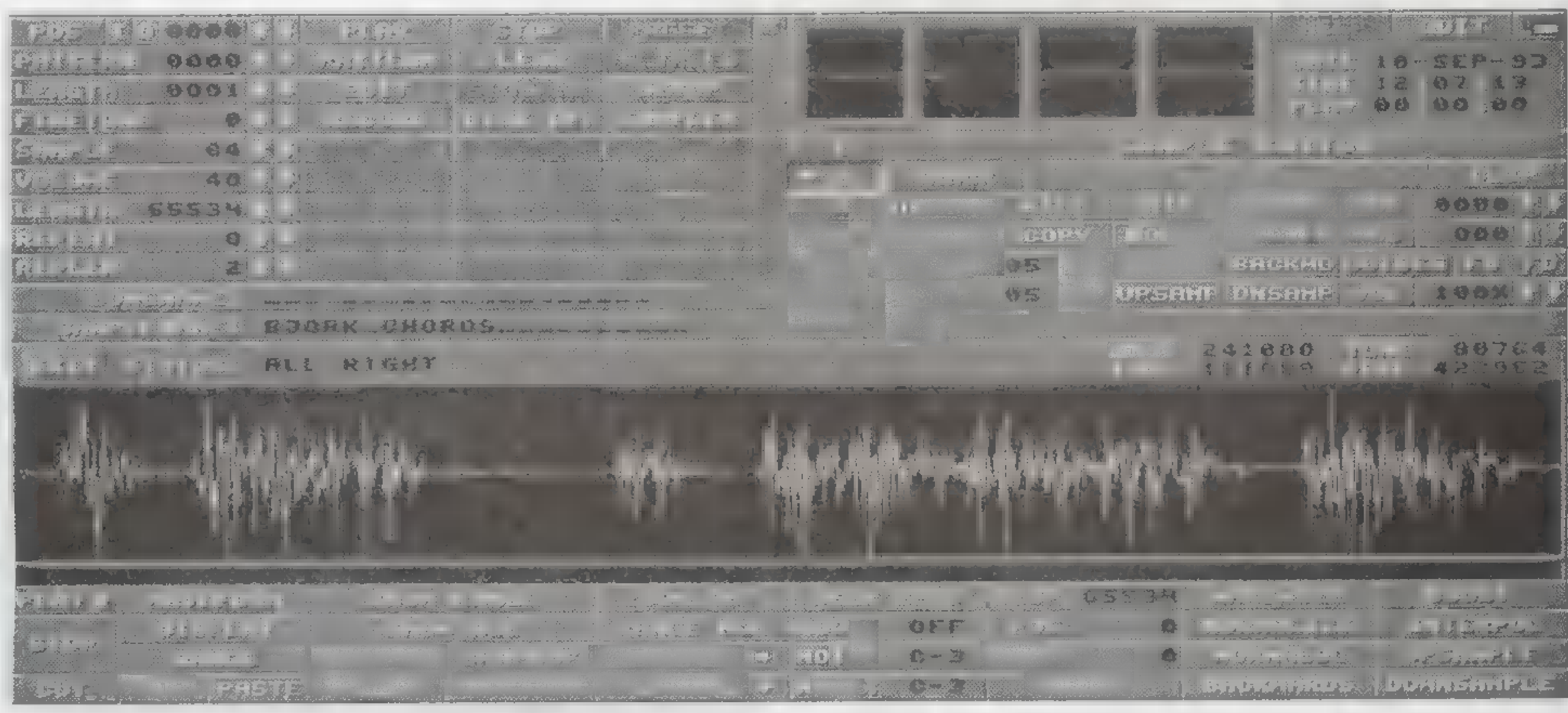
Niesamowicie istotną rzeczą jest właściwy kabel łączący wejście samplera ze źródłem dźwięku. Właściwy - to znaczy możliwie dobrze ekranowany i z pewnie wykonanymi połączeniami. Warto też słyszeć co samplujemy, zatem pomocne mogą być słuchawki. Z mojej praktyki wynika też, że czasami warto zsamplować dźwięk np. z przytłumionymi sopranami, a następnie podbić je programowo. Dźwięk jest wtedy nieco bardziej sztuczny, ale za to czysty i wart nagrania na dysku. Podobnie ma się rzecz z głośnością. Im głośniejszy dźwięk zsamplujemy, tym bardziej będzie on szumiał. Przy dźwiękach samplowanych ciszej poziom szumów jest znikomy, czasami wręcz zerowy. Nie zawsze jednak programowe podgłaśnianie jest najlepszym rozwiązaniem, ponieważ powoduje ono pewne odrealnienie tworzonego dźwięku. Przy próbie wielokrotnego pogłaśniania np. gitary klasycznej lub saksofonu otrzymamy jedynie syntetyczne mutacje tych instrumentów.

Jeżeli szczęśliwym zrządzeniem losu dane nam będzie obcować choć przez parę chwil z syntezatorem, zaraz podłączmy go do samplera i zsamplujmy jak najwięcej akordów. Akordy durowy i mollowy zawsze parami, a poza tym także akordy septymowe i rozszerzone. Następnie proponuję zsamplować dużo instrumentów prowadzących i odpowiednio basy oraz perkusję. Wszelakie efekty typu okrzyki, strzały, kolizje można sobie darować, gdyż nie używa się ich nagminnie (wyłączając techno) i są łatwe do zdobycia w razie potrzeby.

Jeżeli o syntezatorze możemy tylko pomażyć zostaje nam CD lub magnetofon. Wrzucamy więc płytę lub kasetę i samplujemy wszystkie w miarę odseparowane od innych dźwięki. Najczęściej po takiej sesji jesteśmy bogatsi o parę nowych brzmień instrumentów perkusyjnych i gitar. Znacznie trudniej jest wyłapać czyste klawisze, bas, czy wokal, aczkolwiek dla chcącego nie ma nic trudnego. Czasami w wypożyczalniach można znaleźć kompakty z nagranymi brzmieniami, efektami i przeszkadzajkami. Są one przeznaczone dla rozgłośni radiowych (tak mi się wydaje), a brzmienia na nich zawarte są doskonałej jakości. Idealne do zsamplowania.

Bardzo interesujące efekty uzyskuje się przepuszczając swój głos przez tzw. VoiceCodery. Są to jednak urządzenia drogie i dokupowanie ich do Amigi nie miałoby chyba sensu. Zachęcam jednak do eksperymentów z fuzzami od gitar elektrycznych. Efekt jest zdumiewający.

Życzę wszystkim udanych eksperymentów i pełnych dysków sampli. ❑



KUPON KONKURSOWY 1/94

1. Kolejna litera to : . . . .
2. Liczbę dziewięćdziesiąt można zapisać jako : . . . .
3. Komar poleciał : . . . .
4. Największa liczba dyskietek, której nie można kupić to : . . . . .

5. Tekst po rozszyfrowaniu brzmi :
. . . .
. . . .
. . . .

6. Szukaną liczbą pierwszą jest :
. . . .

Imię nazwisko : . . . .
Adres : . . . .
Wiek (nie trzeba wypełniać): . . . .
Zawód (nie trzeba wypełniać): . . . .

1.

KWADRATURA KOŁA

Jarosław Chrostowski

Zajmiemy się dziś konkretnym technicznym problemem - rysowaniem wypełnionych kół, przy okazji poznając podstawy organizacji grafiki (niestety - pobieżnie, po prostu brak miejsca nie pozwala na szersze omówienie tematu). Sprawa wydaje się prosta, wystarczy zaangażować do roboty blitter i mamy po kłopocie.

Zmuszenie blittera do kreślenia wypełnionych kół (koło właściwie jest wypełnione, poprawniej byłoby "wypełnionych okręgów" lub po prostu "kół") nie jest takie proste jakby się na pierwszy rzut oka wydawało. Nie stanowi to na szczęście specjalnego problemu. Diabeł tkwi w szczegółach.

Otóż blitter poprawnie rysuje koła tylko wtedy, gdy koło takie w żadnym miejscu nie wykracza poza obszar mapy bitów. W przypadku, kiedy chcemy narysować koło w jednym z rogów mapy bitów tak, by była widoczna tylko jego ćwiartka, wtedy blitter wypełni nam również obszar poza mapą bitów, w którym znajduje się np. inny program. Krótko mówiąc - czapa. Być może istnieje jakiś sposób, by zmusić blitter do dostrzegania rozmiarów mapy bitów (np. poprzez odwołanie do rastportu), jednak w dostępnej mi literaturze nie znalazłem informacji na ten temat. Jeśli ktoś z czytelników wie coś "w temacie" blitter i koła prosiłbym o kontakt.

Jeśli blitter robi nam takie niespodzianki, wtedy jedynym wyjściem pozostaje samodzielne napisanie procedury rysującej koło. Pierwszy rzut oka na zagadnienie pozwala stwierdzić, że problem rysowania koła sprowadza się do znalezienia par punktów leżących na tej samej wysokości, lecz po obu stronach okręgu, które to pary należy później połączyć linią poziomą. Drugi rzut oka mówi nam, że wystarczy znaleźć punkt leżący w I ćwiartce układu współrzędnych (zakładamy, że środek okręgu znajduje się w środku układu współrzędnych). Wysoka symetria zagadnienia powoduje, że wystarczy odpowiednio pozmieniać (lub nie) znaki poszczególnych wspołrzędnych znalezionego punktu, by otrzymać współrzędne punktów w pozostałych ćwiartkach. Tak więc znalezienie jednego punktu w I ćwiartce powoduje, że prawie natychmiast mamy dwie pary punktów, a więc możemy poprowadzić dwie linie. Trzeci rzut oka mówi, abyśmy zrezygnowali z domorosłej wiedzy typu "pierwiastek sumy kwadratów współrzędnych to promień okręgu". Obliczanie kwadratów jest stosunkowo długie, nie mówiąc o pierwiastkowaniu. Dodatkowe kłopoty mogą sprawiać błędy przybliżeń, które wprowadzą niepotrzebne piksele dające efekt brzydkich zgrubień. Trzeba poradzić sobie w inny sposób.

Tym innym sposobem jest skorzystanie z gotowego algorytmu. Posłużę się algorytmem, który znalazłem w książce "Elementy grafiki komputerowej" Michała Jankowskiego. Rozdział 3.3 na stronie 29 nosi obiecujący tytuł "Rysowanie okręgu", zaś sam algorytm znajduje się na stronie 32. Różnica, czyli 3 strony, to wyprowadzenie algorytmu. Znajomość dowodu nie jest konieczna, by móc skorzystać z podanego schematu, tym niemniej polecam - zakres wiedzy potrzebny do jego zrozumienia nie wykracza poza wiadomości licealistów. Zasada, na której opiera się algorytm, polega na tym, że dla danej pikseli sprawdza się, która z jej sąsiadek najbardziej odpowiada położeniu na rysowanym okręgu. Po wyborze sąsiadki sprawdza się dla niej sąsiednie piksele, itd. Sam algorytm pozwolę sobie przytoczyć w całości:

```
start:
  x=0;   y = R;
  pp = p * p; pp4 = 4 * pp; pp8 = 8 * pp;
  qq = q * q; qq4 = 4 * qq; qq8 = 8 * qq;
  fx = 0; fy = qq8 * R; fs = pp4 - qq4 * R + qq;

  dopóki fx < fy
    {
      x = x + 1;
      fx = fx + pp8;

      jeśli fs <= 0
        to fs = fs + fx + pp4;
      w przeciwnym razie
       {
        y = y - 1;
        fy = fy - qq8;
        fs = fs + fx + pp4 - fy;
       }
    }
  fs = fs - ( fx + fy ) / 2 + 3 * ( pp - qq )

  dopóki y >= 0
    {
      y = y - 1;
      fy = fy - qq8;

      jeśli fs <= 0
        to
          {
            x = x + 1;
            fx = fx + pp8;
            fs = fs + fx - fy + qq4;
          }
        w przeciwnym razie fs = fs - fy + qq4;
    }
```

Użyto następujących oznaczeń: R - promień (liczba naturalna), p, q - liczby naturalne, których iloraz p/q wyznacza aspekt pikseli (proporcje wysokości do szerokości pikseli), x oraz y to współrzędne punktu na okręgu. Założymy, że w naszym przypadku piksele są kwadratowe, a więc p = 1 i q = 1 i użyjmy paru

wyrażeń z języka C. Jeśli dodatkowo pominiemy drugą część algorytmu od linii fs = fs - (fx + fy) / 2 + 3 * (pp - qq) włącznie (sami sprawdźcie, dlaczego można tę część pominąć), wtedy powyższy algorytm upraszcza się do postaci:

```
start:
  x=0; y = R;
  fx = 0; fy = 8 * R; fs = 5 - 4 * R;

  dopóki fx < fy
    {
      fx += 8;

      jeśli fs <= 0
        to fs += fx + 4;
      w przeciwnym razie
       {
         y--;
         fy -= 8;
         fs += fx + 4 - fy;
       }

      x++;
    }
```

Nie ma tu ani grama mnożenia czy pierwiastkowania. Rysowanie okręgu sprowadza się do wykonywania prostych operacji dodawania i odejmowania liczb całkowitych!

Funkcjonowanie algorytmu można sprawdzić przy pomocy załączonego programu. Samo rysowanie kół wykonywane jest przez funkcję DrawGoodCircle, która jako argumentów wymaga: wskaźnika na rastport, na którym będzie rysowane koło, współrzędnych środka koła oraz promienia. Po znalezieniu punktu na okręgu następuje odnalezienie punktu "do pary" i połączenie ich linią. W przypadku przekroczenia wartości zdefiniowanych przez makra MIN_X, MAX_X, MIN_Y i MAX_Y współrzędna punktu przyjmuje wartość makra.

Na koniec polecam przyjrzenie się użytemu mechanizmowi wyprowadzania grafiki. Można po prostu otworzyć ekran i skorzystać z jego rastportu, jednak możemy tu prześledzić poszczególne etapy organizacji grafiki w Amidze. Nadrzędną strukturą jest View, w ramach którego może znajdować się kilka struktur ViewPort. O ile View należy rozumieć jako cały ekran, to ViewPort jest to fragment ekranu, poprzez który widzimy jakąś grafikę. Klasycznym przykładem jest sytuacja, w której uruchamiamy kilka programów otwierających własne ekrany (CED, OPUS, itd.) i zsuwamy je w dół w ten sposób, by wszystkie ekrany były widoczne jednocześnie. Mamy jeden View (widok z pozsuwanymi ekranami), na którym jest kilka ViewPortów (fragmenty poszczególnych ekranów). Tak więc to co widzimy na ekranie jest kontrolowane przez struktury View i ViewPort. To co jest w pamięci to sprawa struktur BitMap i RastPort. O ile ta pierwsza zarządza bezpośrednio planami bitowymi, to ta druga zawiera informacje dodatkowe takie jak kolor pędzla, tryb rysowania, sposób wypełniania. Struktura RastPort jest wykorzystywana przez wszystkie funkcje graficzne systemu Amigi i umożliwia m.in. rysowanie w oknach przysłoniętych przez inne okna.

Rozpoczynamy od zapamiętania wskaźnika na stary View. Wskaźnik ten znajduje się w składowej ActiView struktury GfxBase. Do zainicjalizowania naszej struktury View używamy funkcji systemowej InitView(). Musimy jednakże później uaktualnić wskaźnik na ViewPort tej struktury tak, by wskazywał on na nową strukturę ViewPort. Strukturę tę inicjalizujemy również funkcją systemową, tym razem o nazwie InitVPort(). "Ręcznie" należy podać rozmiary viewportu, na którym będziemy operowali. Zerujemy przy okazji składową Modes. Inicjalizacja składowej RasInfo polega na podaniu wskaźnika do nowej struktury RasInfo. Struktura ta zawiera wskaźnik na mapę bitów oraz zawiera dwie składowe RxOffset i RyOffset, które określają przesunięcie mapy bitów względem viewportu (warto spróbować, co będzie, gdy przydzielimy mapę bitów większą od rozmiarów viewportu i zaczniemy płynnie zmieniać te składowe). Teraz kolej na paletę barw. Funkcja GetColorMap() przydziela i inicjalizuje strukturę ColorMap. Wskaźnik na tę strukturę musi zostać przekazany składowej ColorMap struktury ViewPort. Następnie należy przekopiować paletę barw do tablicy wskazywanej przez składową ColorTable struktury ColorMap. Tablica kolorów zawiera słowa, których ilość odpowiada ilości kolorów w używanej palecie. W każdym słowie 4 bity przeznaczone są na informację o wartości składowej R, podobnie jest dla G i B. Stąd 0xFFF oznacza kolor biały, 0x000 - kolor czarny, 0xF00 - czerwony, 0x0F0 - zielony, 0x00F - niebieski.

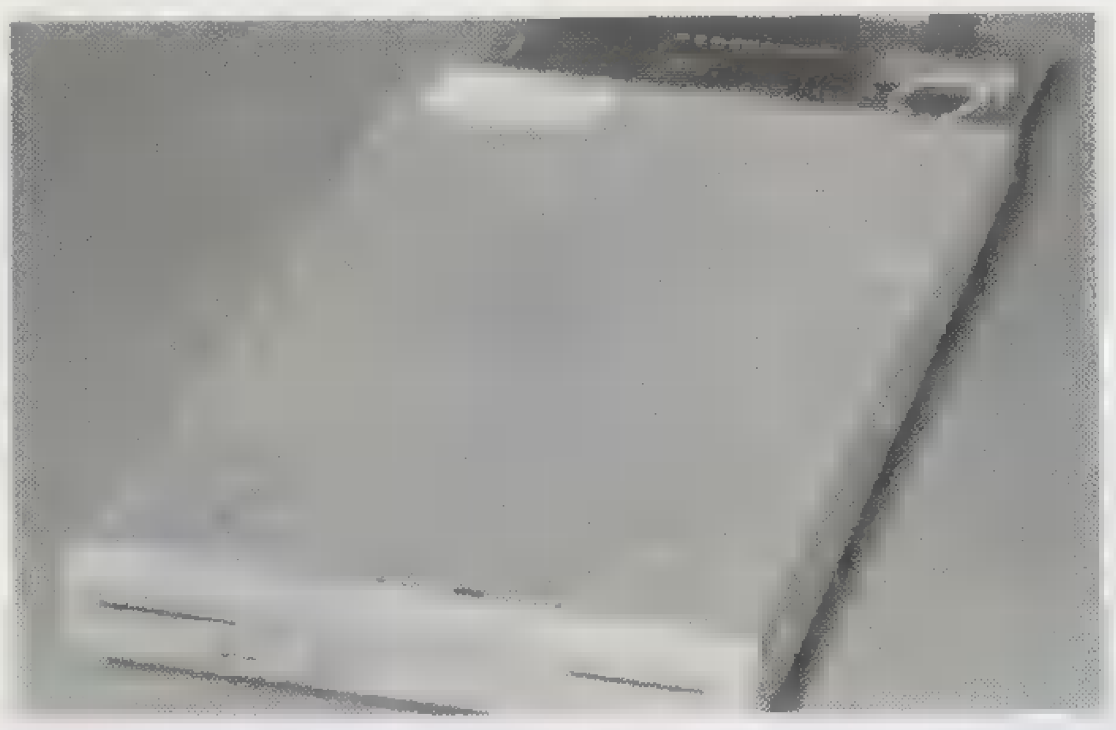
Teraz już będzie z górki. Inicjalizujemy (wreszcie!) mapę bitów przy pomocy (a nie, jak piszą niektórzy: za pomocą) funkcji InitBitMap(), przydzielamy pamięć na poszczególne plany bitowe (dobroczynne działanie funkcji AllocRaster()) i czyścimy każdy plan funkcją BltClear(). Znęcamy się nad strukturą RasInfo: zerujemy składową Next (bo nie ma następnej struktury RasInfo - byłaby w trybie Dual Playfields). Uaktualniamy wskaźnik na mapę bitów BitMap tak, by wskazywał na naszą strukturę. Ustawiamy przesunięcia RxOffset i RyOffset na zero. Zbliżamy się do struktury RastPort. Tu także inicjalizacja wykonywana jest przez funkcję systemową - InitRastPort() i nie ma wiele do roboty. Trzeba tylko wstawić wskaźnik na mapę bitów do składowej BitMap. Całość została przygotowana, teraz trzeba "toto" wyświetlić. W tym celu View i ViewPort należy prowadzić do list coppera, który to koprocesor zajmuje się sprawami ekranu. Załatwiają to funkcje MakeVPort() oraz MrgCop(). Wywołanie funkcji LoadView() pozwoli nam zobaczyć co stworzyliśmy - czyli nic (a było tyle roboty...). Dopiero w tym momencie można trochę porysować.

Przed opuszczeniem programu należy odtworzyć poprzedni View, uaktualnić listy coppera (FreeVPortCopLists() i FreeCprList()), zwolnić przydzielone plany bitowe (FreeRaster()) oraz paletę barw (FreeColorMap()), że o bibliotekach nie wspomnę.

Przyjemnej zabawy. ❑

```
#include <intuition/intuition.h>
#include <graphics/gfxbase.h>
#include <clib/dos_protos.h>
#include <clib/exec_protos.h>
#include <clib/graphics_protos.h>
#include <stdio.h>
#include <stdlib.h>
/* Parametry otwieranego ekranu: szerokość, wysokość, ilość planów. */
/* ilość kolorów. */
#define WIDTH     320
#define HEIGHT    200
#define DEPTH       3
#define COLOURS     8
/* Wskaźniki na używane biblioteki. */
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
/* Wskaźniki na struktury związane z grafiką. */
struct View MyView;
struct View *OldView;
struct ViewPort MyViewPort;
struct RasInfo MyRasInfo;
struct BitMap MyBitMap;
struct RastPort MyRastPort;
/* Paleta kolorów. */
UWORD ColorTable[] = { 0x000,0x800,0xF00,0x080,0x0F0,0x008,0x00F,0xFFF };
/* Deklaracje funkcji. */
void CleanUp( STRPTR message );
void DrawGoodCircle( struct RastPort *rp, WORD offset_x, WORD offset_y,
                     UBYTE radius );
/* To do roboty. */
void main()
{
  UBYTE loop;
  UWORD *pointer;
  /* Otwarcie bibliotek. */
```

```
  IntuitionBase = (struct IntuitionBase *) OpenLibrary( "intuition.library", 0 );
  if( !IntuitionBase )
   CleanUp( (STRPTR) "Nie mogę otworzyć intuition.library!" );
  GfxBase = (struct GfxBase *) OpenLibrary( "graphics.library", 0 );
  if( !GfxBase )
   CleanUp( (STRPTR) "Nie mogę otworzyć graphics.library!" );
  /* Zapamiętanie wskaźnika na stary View. */
   OldView = GfxBase->ActiView;
  /* Inicjalizacja struktury View. */
  InitView( &MyView );
  MyView.ViewPort = &MyViewPort;
  /* Inicjalizacja struktury ViewPort. */
  InitVPort( &MyViewPort );
  MyViewPort.DWidth = WIDTH;
  MyViewPort.DHeight = HEIGHT;
  MyViewPort.RasInfo = &MyRasInfo;
  MyViewPort.Modes = 0;
  /* Przydzielenie mapy kolorów. */
  MyViewPort.ColorMap = (struct ColorMap *) GetColorMap( COLOURS );
  if( MyViewPort.ColorMap == NULL )
   CleanUp( (STRPTR) "Nie mogę przydzielić mapy kolorów!" );
  /* Ustawienie wartości poszczególnych kolorów. */
  pointer = (UWORD *) MyViewPort.ColorMap->ColorTable;
  for( loop=0; loop<COLOURS; loop++ )
    *pointer++ = ColorTable[ loop ];
  /* Inicjalizacja mapy bitów. */
  InitBitMap( &MyBitMap, DEPTH, WIDTH, HEIGHT );
  /* Przydział pamięci na poszczególne plany bitowe i zerowanie planów. */
  for( loop=0; loop<DEPTH; loop++ )
    {
      MyBitMap.Planes[ loop ] = (PLANEPTR) AllocRaster( WIDTH,
                                                  HEIGHT );
      if( MyBitMap.Planes[ loop ] == NULL )
       CleanUp( (STRPTR) "Brak pamięci na raster!" );
      BltClear( MyBitMap.Planes[ loop ], RASSIZE( WIDTH, HEIGHT ), 0 );
    }
   /* Inicjalizacja struktury RasInfo. */
  MyRasInfo.BitMap = &MyBitMap;
  MyRasInfo.RxOffset = 0;
  MyRasInfo.RyOffset = 0;
  MyRasInfo.Next = NULL;
   /* Inicjalizacja rastportu. */
  InitRastPort( &MyRastPort );
  MyRastPort.BitMap = &MyBitMap;
   /* Wyświetlenie naszego view. */
  MakeVPort( &MyView, &MyViewPort );
  MrgCop( &MyView );
  LoadView( &MyView );
   /* Rysowanie kolejnych okręgów. */
  SetDrMd( &MyRastPort, JAM1 );
  for ( loop=0; loop<100; loop++ )
    {
      SetAPen( &MyRastPort, rand() % (COLOURS-1) + 1 );
      DrawGoodCircle( &MyRastPort, rand() % (320-1) + 1,
                      rand() % (200-1) + 1, rand() % (200-1) + 1 );
    }
   /* Chwila zwłoki. */
  Delay(200);
   /* Odtworzenie starego view i zakończenie programu. */
  LoadView( OldView );
  CleanUp( (STRPTR) "KONIEC!!!" );
}
#define MIN_X   0
#define MAX_X  319
#define MIN_Y   0
#define MAX_Y  199
/* Funkcja rysuje na rastporcie wskazywanym przez rp koło o        */
/* promieniu     radius o środku na współrzędnych offset_x,        */
/* offset_y. Wartości makr MIN_X, MAX_X, MIN_Y, MAX_Y    */
/* określają rozmiary ekranu, poza którymi koło nie jest rysowane. */
void DrawGoodCircle( struct RastPort *rp, WORD offset_x, WORD offset_y,
                     UBYTE radius )
{
  UBYTE loop;
  WORD x, y, x1, x2, y1, y2, fx, fy, fs;
  /* Nic dodać, nic ująć - po prostu przepisany na C algorytm kreślenia kół. */
  x = 0;
  y = radius;
  fx = 0;
  fy = 8 * radius;
  fs = 5 - 4 * radius;
  while( fx < fy )
    {
      fx += 8;
      if ( fs <= 0 )
        fs += fx + 4;
      else
        {
          y--;
          fy -= 8;
          fs += fx + 4 - fy;
        }
      for ( loop=0; loop<2; loop++ )
        {
          /* Rysowanie linii. */
          if ( loop )
            {
              x1 = y + offset_x;
              x2 = offset_x - y;
              y1 = x + offset_y;
              y2 = offset_y - x;
            }
          else
            {
              x1 = x + offset_x;
              x2 = offset_x - x;
              y1 = y + offset_y;
              y2 = offset_y - y;
            }
          if ( y1 >= MIN_Y && y1 <= MAX_Y )
            if ( x2 < MAX_X && x1 > MIN_X )
              {
                if ( x2 < MIN_X )
                  x2 = MIN_X;
                if ( x1 > MAX_X )
                  x1 = MAX_X;
                Move( rp, x2, y1 );
                Draw( rp, x1, y1 );
              }
          if ( y2 >= MIN_Y && y2 <= MAX_Y )
            if ( x2 < MAX_X && x1 > MIN_X )
              {
                if ( x2 < MIN_X )
                  x2 = MIN_X;
                if ( x1 > MAX_X )
                  x1 = MAX_X;
                Move( rp, x2, y2 );
                Draw ( rp, x1, y2 );
              }
        }
      x++;
    }
}
/* Funkcja zwalnia wszystko, co jest do zwolnienia, drukuje tekst */
/* wskazywany przez message i kończy działanie programu.        */
void CleanUp( STRPTR message )
{
  UBYTE loop;
   /* Usunięcie view z list coppera. */
  FreeVPortCopLists( &MyViewPort );
  FreeCprList( MyView.LOFCprList );
   /* Zwolnienie planów bitowych. */
  for( loop=0; loop<DEPTH; loop++ )
    if( MyBitMap.Planes[ loop ] )
      FreeRaster( MyBitMap.Planes[ loop ], WIDTH, HEIGHT );
   /* Zwolnienie mapy kolorów. */
  if( MyViewPort.ColorMap )
    FreeColorMap( MyViewPort.ColorMap );
   /* Zamknięcie bibliotek. */
  if( GfxBase )
    CloseLibrary( (struct Library *) GfxBase );
  if( IntuitionBase )
    CloseLibrary( (struct Library *) IntuitionBase );
  printf( "%s", message );
  exit(0);
}
```



producent: Power Computing,
dystrybutor: Power Computing,
Unit 8 Railton Road, Woburn Road Industrial Estate, Kempston, Bedford MK42 7Pn,
tel. +0234 843388, fax. +0234 840234
cena: 76 funtów (wersja dla A4000 80 funtów)
zgodność: 3.0 i wyższe

Wewnętrzny dysk o dużej gęstości (Stacja XL)

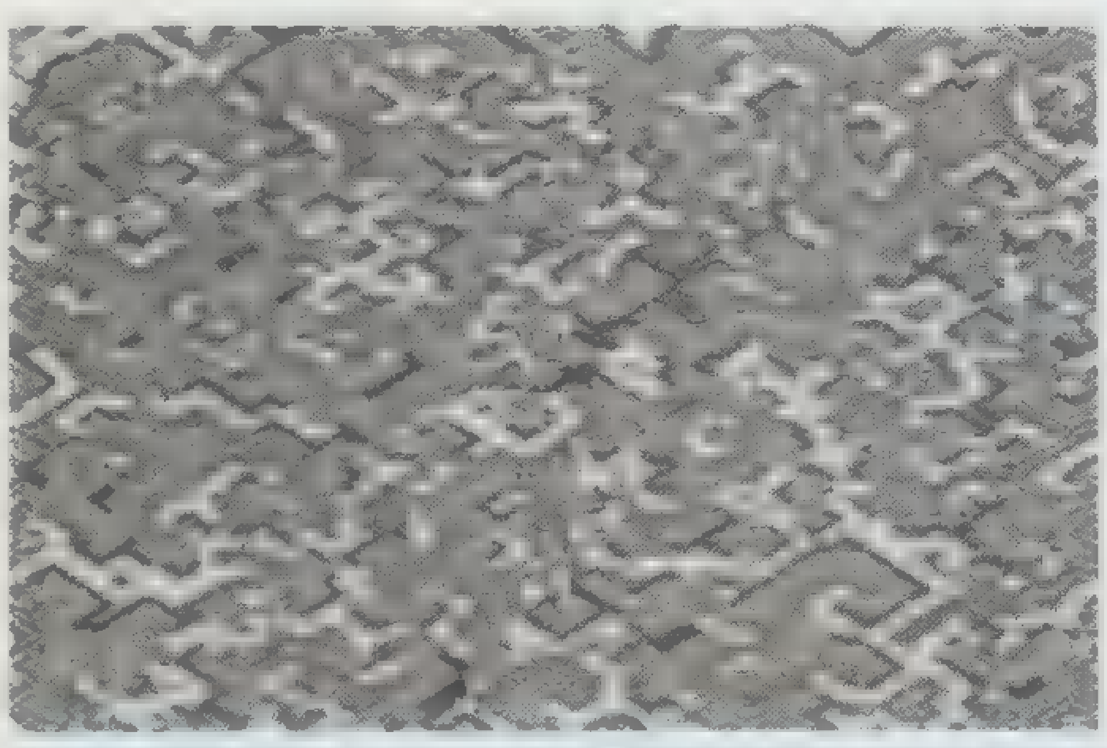
Wielkie ilości danych jakie przetwarza Amiga dają się szczególnie odczuć, gdy musimy je magazynować lub przenosić z jednego komputera na drugi. Rozwiązaniem może być oczywiście zewnętrzna stacja o dużej gęstości zapisu, którą opisaliśmy w świątecznym numerze AMIGOWCA. Dla niektórych praktyczniejszym rozwiązaniem będzie zakup stacji wewnętrznej o takich samych możliwościach. Montaż stacji jest bardzo prosty - nie należy jedynie pomylić ułożenia dwóch wtyków i to wszystko (no, oczywiście należy stację przykręcić śrubami mocującymi). Stacja ma możliwość czytania i formatowania dysków o pojemności 1.76 MB, jak i normalnych dysków 880 KB. Niestety jest jedno "ale". Zanim zaczniemy pracę z nową stacją musimy wprowadzić pewien programik, który będzie uruchamiał się ze Startup-Sequence (pliku początkowego). Wychodzi na to, że najpierw trzeba zainicjalizować system z małej dyskietki lub twardego dysku, a dopiero potem będzie można pracować z "dużymi" dyskami. Równie niejasno określona przez producenta jest możliwość współpracy z Amigami różnych typów. O ile przy Amidze 1200 nie powinno być z tym żadnych problemów, o tyle gorąco radzimy posiadaczom innych wersji Amig sprawdzić kompatybilność omawianej stacji. Stacja ta jest dostępna w wersji specjalnej do Amig 4000.



producent: MegaGeM
dystrybutor: MegaGeM,
1903 Adria, Santa Maria,
CA 93454,
USA
tel. 0101 805 3491104
cena: 65$
zgodność: 2.0 i wyższe

Proteza dla Visty (Scape Maker)

Wszyscy użytkownicy ViPro (program opisywany w AMIGOWCU 9-10/93) zdążyli zapewne zorientować się, że w programie nie ma możliwości wmontowywania zwykłych rysunków w krajobrazy. Wymagałoby to bowiem ich przetworzenia w standard używany przez Vistę (pliki z dopiskiem .DEM). Dla wypełnienia tej "dziury" powstał program ScapeMaker. Umożliwia on właśnie przetwarzanie zwykłych grafik w formacie IFF na format Visty Pro. Grafiki mogą być przetwarzane w całości lub we fragmentach, a po przetworzeniu mogą być wykorzystywane oczywiście tylko przy użyciu Visty. Program pozwala także na edycję już istniejących plików .DEM i dokonywania w nich pewnych poprawek - jak np. wygładzanie. Istnieje możliwość wkomponowywania zwykłych grafik we fraktalowe krajobrazy. Efektów jakie mogą powstać dzięki temu programowi nie powstydzą się profesjonalni twórcy grafiki. To po prostu zapiera dech w piersiach. Oczywiście nie wszystkie grafiki wyjdą tak, jak byśmy??? sobie tego życzyli. Niektóre trzeba specjalnie dostosować, co wymaga nieco więcej trudu, inne natomiast mogą wydawać się nieco zmienione po wczytaniu ich do Visty. Wielu domorosłych twórców może jednak wykorzystać ten program oraz Vistę Pro do tworzenia profesjonalnych filmów. Gorąco zachęcamy.



producent: MegaGem
dystrybutor: MegaGem,
1903 Adria Santa Maria,
Ca 93454,
USA
tel. 0101 805 349 1104
zgodność: 2.0 i wyższe
cena: 90$

Rośnij komórko (CellPro)

Pewnie tylko część z Was wie o co chodzi w emulacjach komórkowego wzrostu świata. Emulacje tego typu były swego czasu opisywane w "Problemach", a pierwszym programem umożliwiającym ich przeprowadzanie na komputerach był program Life. Program ten umożliwiał prześledzenie symulacji życia prostych organizmów jednokomórkowych w ściśle kontrolowanych warunkach. I tak każda komórka mogła mieć do 8 sąsiadów. Ilość sąsiadów warunkowała zachowanie się komórki. Gdy było ich zbyt wielu, komórka po prostu ginęła. Podobnie działo się przy zbyt małej ilości sąsiadów. Oczywiście, gdy jakieś pole było otoczone tylko przez dwie komórki to mogło w nim powstać nowe życie. Dziecinada. A jednak nie. Symulacje tego typu umożliwiają prześledzenie zmian jakie zachodzą w środowisku w zależności od stanu wyjściowego i czynników zewnętrznych. Programy te podbudowane pewną dozą wiedzy pozwalają przewidywać to, co nieprzewidywalne. Np. wzrost kolonii bakteryjnych. A stąd już tylko kroczek do zastosowań praktycznych: np. oceny działania antybiotyków i innych leków. Wersja programu Life na Amigę to właśnie CellPro. Dzięki temu prostemu programowi oraz modyfikacjom jakie możemy sami do niego wprowadzać, młodzi adepci "nauk tajemnych" już w szkołach mogą zbliżyć się bardziej niż teoretycznie do zrozumienia pojęcia "życie".

Pokwitowanie dla wpłacającego

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla posiadacza rachunku

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla BANKU

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

AMI-MARKET

Wystarczy wypełnić DRUKOWANYMI literami zamieszczony obok blankiet. Ogłoszenia mogą zamieszczać tylko osoby prywatne. Ważny jest tylko oryginalny, wycięty z czasopisma blankiet (NIE jego kserokopia!). Ogłoszenia dotyczące sprzedaży muszą zawierać cenę. Uwaga! Prosimy zaznaczyć czasopismo, w którym ma się ukazać ogłoszenie:

p **AMIGOWIEC**

p **Świat Gier Komputerowych**

ZAPRASZAMY

imię nazwisko

miejsce zamieszkania

kod miejscowy

telefon, fax

UWAGI

BŁYSKAWICZNE DARMOWE OGŁOSZENIA sprzedam, kupię, wymienię

Pokwitowanie dla wpłacającego

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla posiadacza rachunku

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla BANKU

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla BANKU

AMIGOWIEC

Zamówienie na prenumeratę i numery archiwalne

Zamawiam prenumeratę AMIGOWCA od numeru:

Zamawiam prenumeratę Dysków P.D. od numeru:

Zamówienie na numery archiwalne:

5/92	6/92	12/93	5-6/93	7-8/93	9-10/93
1/94					

Zamówienie na dyskietki Public Domain:

1	2	3	4	5	6
7	8	9	10	11	12
A	B	C	1/93	2/93	3-4/93
5-6/93	7-8/93	9-10/93	1/94		

Odcinek dla posiadacza rachunku

AMIGOWIEC

Zamówienie na prenumeratę i numery archiwalne

Zamawiam prenumeratę AMIGOWCA od numeru:

Zamawiam prenumeratę Dysków P.D. od numeru:

Zamówienie na numery archiwalne:

5/92	6/92	12/92	5-6/93	7-8/93	9-10/93
1/94					

Zamówienie na dyskietki Public Domain:

1	2	3	4	5	6
7	8	9	10	11	12
A	B	C	1/92	2/93	3-4/93
5-6/93	7-8/93	9-10/93	1/94		

Pokwitowanie dla wpłacającego

AMIGOWIEC

Zamówienie na prenumeratę i numery archiwalne

1. Aby zaprenumerować AMIGOWCA i / lub dyskietki Public Domain należy wpłacić na nasze konto:
- 138 tys. za 6 wydań AMIGOWCA od dowolnie wybranego numeru.
- 228 tys. za 6 dyskietek Public Domain od dowolnie wybranego numeru.

Aby otrzymać numery archiwalne AMIGOWCA i / lub zamówić dyskietki PD wystarczy wpłacić na nasze konto:
- 7 tys. za numery (5/92 i 6/92),
- 16 tys. za numery (12/92),
- 18 tys. za numery (5-6/93, 7-8/93),
- 23 tys. za numer (9-10/93, 1/94)
- 48 tys. za każdą dyskietkę PD,

2. Podać czytelnie swój adres.
3. Zaznaczyć na obu odcinkach blankietu, które numery zamawiamy.

W razie niejasności, czy jakichkolwiek opóźnień prosimy o informacje na adres:

P.W.H. "ALFIN" sp. z o.o.
tel. 28-79-20, fax. 22-64-03
ul. Świętojańska 2/7, 85-017 Bydgoszcz

Drodzy Czytelnicy!

Proponujemy półroczną prenumeratę naszych czasopism: AMIGOWIEC i ŚWIAT GIER KOMPUTEROWYCH rozpoczynającą się od dowolnie wybranego numeru. Można również zaprenumerować dyskietki Public Domain oraz zamówić archiwalne egzemplarze AMIGOWCA.

Wystarczy czytelnie wypełnić obie strony wybranego blankietu prenumeraty i wpłacić odpowiednią kwotę na podane konto. Świat Gier Komputerowych w prenumeracie kosztuje taniej - tylko 23 tys. zł. Również AMIGOWIEC jest tańszy w prenumeracie - tylko 23 tys. zł. Dyskietki Public Domain są tańsze dla prenumeratorów naszych czasopism - tylko 38 tys. zł.

Odcinek dla BANKU

Świat GIER KOMPUTEROWYCH

od numeru: | od numeru:

5-6/93	7-8/93	9-10/93	11-12/93	1/94	

Odcinek dla posiadacza rachunku

Świat GIER KOMPUTEROWYCH

Zamawiam prenumeratę

od numeru: | od numeru:

Numery archiwalne

5-6/93	7-8/93	9-10/93	11-12/93	1/94	

Pokwitowanie dla wpłacającego

Świat GIER KOMPUTEROWYCH

Aby zaprenumerować Świat Gier Komputerowych, należy:

1. Wybrać rodzaj prenumeraty i zaznaczyć, od którego numeru ma się rozpocząć.
2. Podać czytelnie swój adres.
3. Wpłacić odpowiednią kwotę na nasze konto, tj. 138.000,- za 6 wydań lub 276.000,- za 12 wydań.

Jednocześnie informujemy, że P.W.H. "ALFIN" ani ŚGK nie będą odpowiadać za problemy wynikłe z błędnego wypełnienia blankietu.

W razie niejasności, czy jakichkolwiek opóźnień prosimy o zgłaszanie się na adres:

P.W.H. "ALFIN" sp. z o.o.
tel. 28-79-20, fax 22-64-03
ul. Świętojańska 2/7,
85-017 Bydgoszcz

AMI-MARKET

Sprzedam C64II, magnetofon, Black Box v.4, osłona klawiatury oraz 150 gier. Cena 1.4 mln zł. **Bernard Błaszczak, ul. Bałkańska 9/36, 85-167 Bydgoszcz, tel. 71-64-12.**

Sprzedam Amigę 2000D (kickstart 2.04), 3MB RAM. Cena 7 mln zł. **Krzysztof Wirszyłło, Bydgoszcz tel. 22-45-69 lub 28-79-20 (praca).**

Kupię drukarkę do A500. Oferty z ceną i wydrukiem proszę kierować na adres: **Artur Marciniak, ul. Zwycięstwa 34, 72-410 Golczewo, 447.**

Sprzedam Amigę 500 wraz z całym oprzyrządowaniem (monitor kolor, stacja 5.25", RAM 512, 400 dysków, prasa). Cena: 10 mln zł. Bardzo dobry stan. **Artur Drzymała, ul. Nowotki 7, 32-513 Jaworzno.**

Kupię twardy dysk 120 MB do Amigi 1200. Proszę o oferty z ceną. **Tomasz Laska. ul. Prądzyńskiego 6b/11, 87-100 Toruń, tel. 16-22-27.**

Kupię programy przetwarzające i akcesoryjne wraz z opisem dla systemu AMOS. **Włodzimierz Gut, ul. Kościuszki 232, 42-583 Bobrowniki.**

Wymienię doświadczenia z użytkownikami Amigi 500 (mam olbrzymią programotekę, instrukcje, literaturę). **Andrzej Orechwa, ul. Ogrodowa 11/123, 00-893 Warszawa.**

Klub użytkowników AMOSA! Aby uzyskać informacje prześlij kopertę z Twoim adresem i naklejonym znaczkiem. **Jarosław Lang, oś. XXV-Lecia 31/119, 58-260 Bielawa.**

Sprzedam numery "ŚGK" 1-10/93. Cena 135 tys. zł. Numery "Amigowca" 1-8/93. Cena 80 tys. zł. Koszty przesyłki wliczone. **Sławomir Łyszkiewicz, ul. Obrońców Helu 10b/2, 84-150 Hel, tel. 750-588 (po godz. 15).**

Zamienię CPC464+monitor kolor, konsolę Osaka, Rometa (wszystko w idealnym stanie) na Amigę 500 z monitorem. **Przemysław Przybylski, Lipie 45, 42-165 Lipie.**

Kupię Amigę 600. **Maciej Gołębiowski, ul. Romera 2/172, 76-200 Słupsk, tel. 444-779.**

Poszukuję "Amigi" nr 2/92; kupię lub wymienię. **Czesław Czyżyk, ul. 1-ego Maja 1/3, 58-520 Janowice Wielkie.**

Przegram gry i programy na Amigę. Cena 6 tys. za przegranie 1 dysku. Duży wybór. Nowości! **Krzysztof Żymła, ul. Halbera 28, 44-200 Rybnik, tel. 23-401.**

Kupię Amigę 500+ za 3-4 mln zł. W bardzo dobrym stanie. Uwaga: gdy będzie jeszcze na gwarancji, dam więcej. **Maciej Wesołowski, ul. Kazimierza Wielkiego 1a, 41-902 Bytom.**

Sprzedam programy: na Amigę za 10 tys. zł, na IBM za 15 tys. zł. Katalog gratis. Dyskietki 5.25" Maxwell - za 12 tys. zł. **Arek Klimek, ul. Jana Pawła 18/3, 14-100 Ostróda, tel. 23-52.**

Sprzedam komputer A600+pokrywa, 2 joysticki. Gwarancja (09.94). Cena: ok. 5.5 mln zł lub wymienię na A500+. **Krzysztof Siejak, ul. Przemysłowa 47/8, 61-541 Poznań.**

Sprzedam A1200 (gwarancja 04.94). Cena: 8.5 mln zł. Atari 800XL i magnetofon 12XC. Cena: 0.8 mln zł. **Marek Marciniak, ul. K. Wielkiego 8g/15, 47-220 Kędzierzyn Koźle.**

Sprzedam dysk twardy 120 MB, kontroler, 4 MB RAM do Amigi 500. Cena ok. 9.5 mln zł. **Rafał Rudzki, ul. Brzozowa 52/12, 41-515 Chorzów, tel. 464-813.**

Sprzedam dyskietki z programami do Amigi po 10 tys. zł i Action Replay MkIII do Amigi (z gwarancją) za 1.4 mln zł. **Jacek Gierczycki, ul. Komorowskiego 11, 48-200 Prudnik (woj. opolskie), tel. 626-48.**

Poszukuję opisu po polsku programu na Amigę 500+: ProTracker 2.3A. **Zbigniew Styrcz, ul. Żeromskiego 95/97m66, 26-600 Ra**

Sprzedam literaturę z serii mikrokomputery: "Basic dla początkujących", "Basic dla wszystkich" i "Commodore C64". **Michał Dargacz, ul. Rejtana 15/40, 84-200 Wejherowo, tel. 72-02-34.**

Wymiana programów na A500. Dużo nowości. Na poważne oferty - gwarancja odpowiedzi. **Jarosław Żółtaszek, ul. Zapolskiej 61/9, 43-135 Tychy, tel. 119-70-93.**

Sprzedam Amigę 2000 3 MB RAM, Quantum 80 MB SCSI, dwie stacje 3,5", emulator IBM AT - 1 MB RAM, stacja 5,25" 1,44 MB. **Jarosław Chrostowski, ul. Chałubińskiego 1/67, 85-792 Bydgoszcz, tel. 43-76-49.**



POTĘGA MOŻLIWOŚCI - ZAUFAJ DOŚWIADCZENIU !

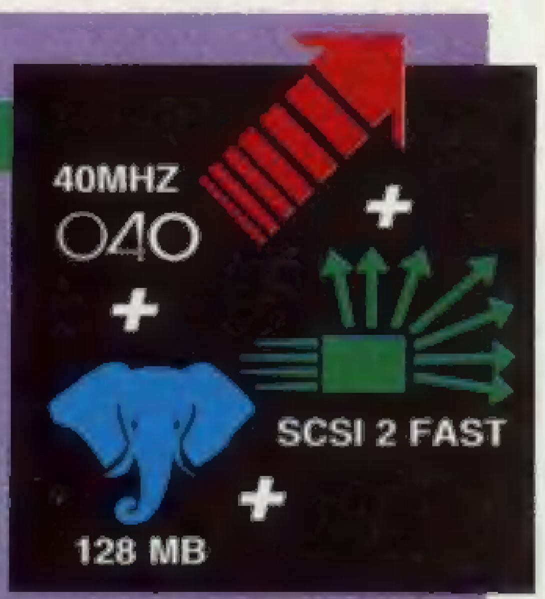
GVP to najlepszy wybór dla posiadacza każdej Amigi

GVP
GREAT VALLEY PRODUCTS, INC.
USA

Firma Great Valley Products jest liderem w dziedzinie peryferii i rozszerzeń do Amigi od 1988 roku.

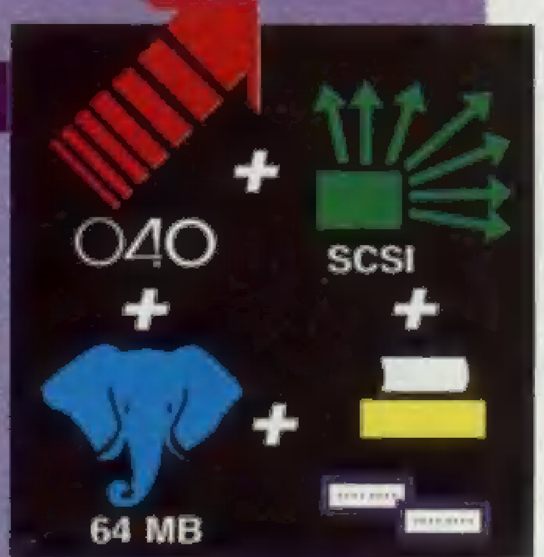
TYLKO RZECZY NAJLEPSZE SĄ WYSTARCZAJĄCO DOBRE !
KUPUJESZ DROŻEJ - WYCHODZI TANIEJ !

GVP
GREAT VALLEY PRODUCTS, INC.
USA

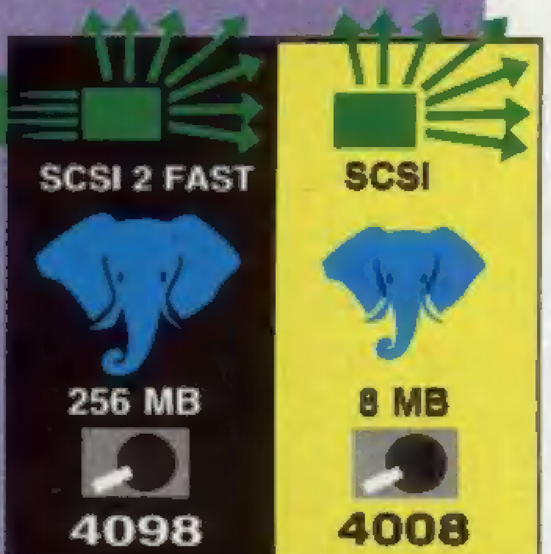


A4000 G-Force'040
Jeśli jesteś profesjonalistą i czujesz się skrępowany niewystarczającą prędkością Twojej Amigi 4000 lub 3000, karta ta sprawi, że zapomnisz o tym problemie na całe lata. A4000 GForce'040 to m.in.: zegar 40MHz, do 128MB 32–bitowej bardzo szybkiej pamięci, gniazdo rozszerzeń na przykład dla opcjonalnego modułu SCSI–II FAST. Karta pracuje z oboma typami A4000, A4000T (z procesorami 030 lub 040).

G-FORCE



G-Force '040/33 Combo
Karta Combo rozbudowana do absolutnego maksimum. Twoje programy będą frunąć przy olbrzymiej mocy pracującego z częstotliwością 33MHz procesora 68040. Możesz tej mocy dostarczyć do 64MB 32–bitowej pamięci.
Oczywiście nasz interfejs SCSI II znajduje się na karcie, dodajemy także jako bonus dodatkowe porty: równoległy i buforowany, szybki port szeregowy.



4008 SCSI II / 4098 SCSI2FAST
Sięgnij do świata SCSI przy pomocy jednej z tych prostych w instalacji kart. Natychmiast zyskasz dostęp do tysięcy urządzeń, takich jak twarde dyski, napędy SyQuest, CD–ROM.
Możesz dołączyć do swojej Amigi 4000 do siedmiu takich urządzeń. Karta pracuje także z Amigą 2000 i umożliwia rozszerzenie pamięci RAM o 8MB. Karta 4098 wyposażona jest w pełny 32–bitowy interfejs, pozwalający na osiągnięcie niewiarygodnej maksymalnej szybkości przesyłu do 10MB/s, a dzięki miejscom na osiem modułów SIMM, możliwe jest rozszerzenie pamięci do 256MB! Dzięki niewielkim rozmiarom karty (technologia montażu powierzchniowego), możliwe jest zamontowanie dowolnego 3.5" napędu bezpośrednio na karcie, zapewniając maksymalną wygodę. Przekonaj się o różnicy.



IV-24 2.0 -IMPACT VISION 24- Profesjonalny Genlock
To jest to, czego szukałeś, potrzebując profesjonalnej jakości genlocka do Twojej Amigi 2000, 3000 lub 4000. Wysoko zintegrowane układy dają najlepszy, najczystszy obraz. Przy możliwości pracy na wyjściu i wejściu z sygnałami RGB, composite video, SVHS, component (Betacam oraz M–II), przy 24–bitowym frame bufferze i pracującym w czasie rzeczywistym digitizerze obrazu, IV24 jest genlockiem potrzebnym każdemu profesjonaliście.
Dodawane oprogramowanie uzupełnia obraz Profesjonalnego Genlocka.

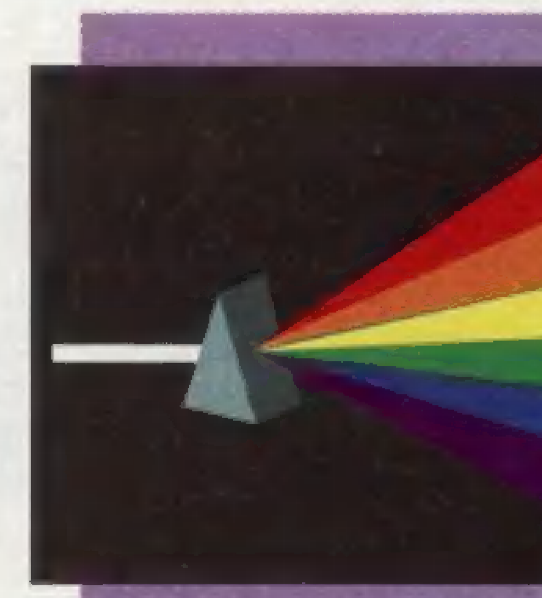
TBC plus



TBC Plus
Ten profesjonalny, cyfrowy Time Base Corrector (Korektor Bazy Czasowej) pracuje w najwyższej jakości, standardzie 4:2:2 (8 bitów).
PLUS: karta zawiera także pracujący w czasie rzeczywistym frame–grabber oraz 24–bitowy frame–buffer. PLUS: karta potrafi odbierać i generować kody czasowe SMPTE/EBU. PLUS: to niesamowite urządzenie transkoduje sygnały PAL–NTSC–SECAM, zarówno composite video jak i Y/C.
PLUS: możliwość wyboru jednego z trzech kanałów video (composite video i Y/C).
PLUS: programowalne efekty specjalne.

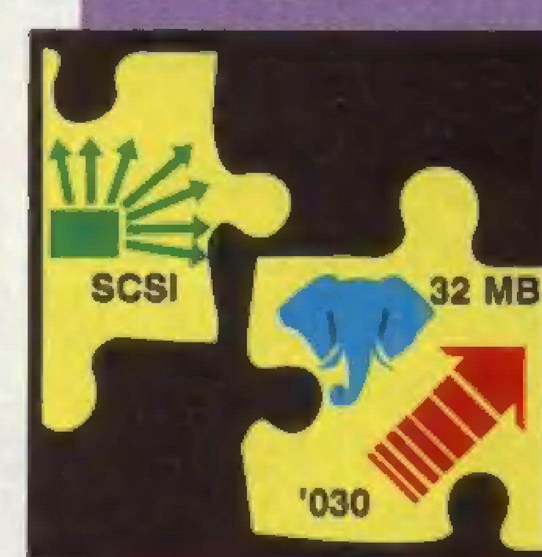


G-Lock
Możesz łączyć w jedną całość obraz z video, dźwięk oraz grafikę Amigi przy pomocy G-Lock'a i dowolnej Amigi! Pozbędziesz się problemów dzięki wbudowanemu transkoderowi pomiędzy Composite Video, Y/C, RGB oraz YUV! Pełne wykorzystanie możliwości kości AGA, jak również także "zwykłych" Amig 500, 2000 i 3000. Doskonały programowy interfejs czyni łatwym sterowanie G-Lockiem. Użytkownicy programu Scala mają do dyspozycji moduł EX przeznaczony dla G-Lock'a, pozwalający im na użycie go w ich prezentacjach. Dodaj jeszcze możliwość obróbki dwukanałowego dźwięku, a będzie jasne, że G-Lock to najlepszy wybór dla każdego użytkownika Amigi.



SPECTRUM

EGS 28/24 SPECTRUM
Zostaw w tyle kości AGA, kupując tą pracującą w czasie rzeczywistym, 24–bitową kartę graficzną. Programowalne rozdzielczości z maksymalną 1600x1280 punktów! 800x600 punktów w 24 bitach! Do karty dodajemy specjalny kabel przelotowy, umożliwiający pracę z jednym monitorem. Wiele programów już w tej chwili pracuje z kartą SPECTRUM, dodajemy też gratis doskonały program EGS Paint. Daj swojej Amidze możliwości graficzne stacji roboczej już dziś i patrz, co traciłeś!



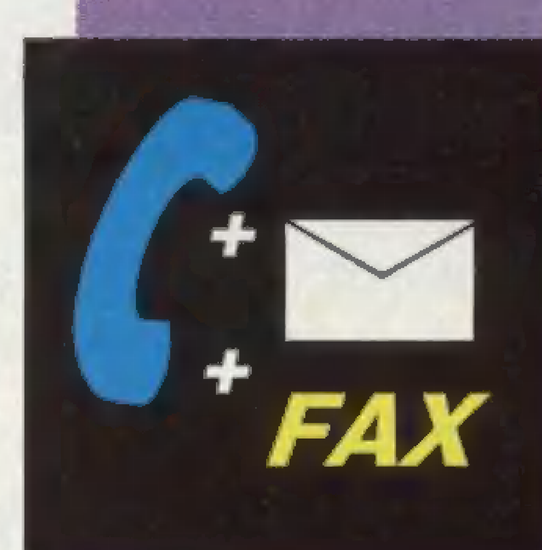
A1230 TURBO+ Series II
Przy zegarze procesora 50MHz, opcjonalnie 40 MHz, możesz być właścicielem najszybszej Amigi 1200 na świecie! Już dziś możesz mieć do 32MB 32–bitowej pamięci. Z pracującym z zegarem 50MHz/40MHz koprocesorem, twoje obliczenia zmiennoprzecinkowe nigdy nie były szybsze. Prosta instalacja w złączu pod spodem A1200 nigdy nie narusza gwarancji, a przy kartach Series II zyskujesz uniwersalność naszego dodatkowego gniazda rozszerzeń. Kupując A1291 SCSI Kit możesz wyposażyć swoją A1200 w najszybszy interfejs SCSI (wyposażony w gniazdo zewnętrzne, na przykład do CD–ROM). Po prostu podłączasz go do gniazda karty. Inne rozszerzenia to między innymi 16–bitowa karta dźwiękowa z samplerem i frame–grabber (digitizer obrazu video pracujący w czasie rzeczywistym).



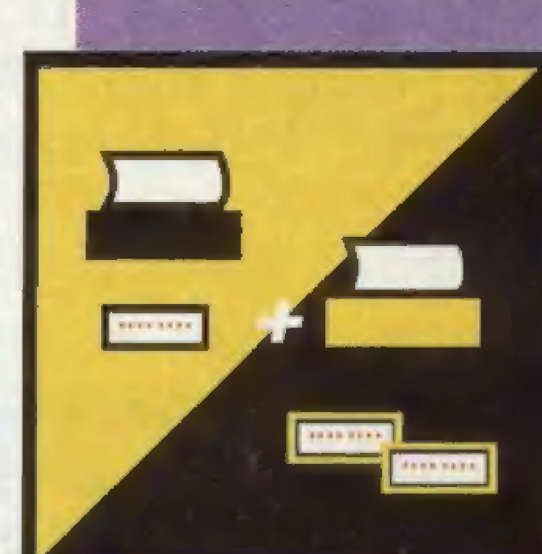
DSS8+ DIGITAL SOUND STUDIO

DSS8+ Po prostu genialne! To najmniej zakłócający i najbardziej profesjonalny sampler (digitizer dźwięku) dotychczas skonstruowany. Zrobiony z odpornego na uderzenia, przezroczystego tworzywa, jest to sampler do Amigi, który musisz mieć. Dołączany program Digital Sound Studio pozwala na samplowanie, edycję, komponowanie, odtwarzanie dźwięku w stereo, jak również na zapis gotowych utworów w formacie .mod (Tracker).

PHONE PAK VFX



PhonePak VFX 2.0
Jeśli chcesz użyć systemu VoiceMail, naciśnij 1. Jeśli chcesz wysłać fax, naciśnij 2. Jeśli jeszcze chcesz, by wszystkie funkcje były zautomatyzowane, realizowane w zaprogramowanym czasie, i abyś był powiadamiany, gdy jest coś do ciebie, kup PhonePak VFX 2.0 dziś! Pełna integracja, możliwość użytkowania nieograniczonej liczby "skrzynek pocztowych" i odbioru prywatnych faxów. Możesz tworzyć faxy z wydruków generowanych przez dowolny program. Możesz zadzwonić z innego miejsca i przesłać wcześniejsze faxy. Faxowanie z i bez użycia papieru.



ioExtender
Czujesz się ograniczony brakiem możliwości? Pozwól firmie GVP wyzwolić Cię przy pomocy karty ioExtender. Na jednej karcie znajdziesz dodatkowy port równoległy, pozwalający na jednoczesne podłączenie drukarki i digitizera (jak np. DSS8+). Nigdy więcej kłopotliwych zewnętrznych przełączników. Na karcie znajdują się także dwa (tak, dwa!) bardzo szybkie, buforowane porty szeregowe. Nigdy więcej straconych danych, nigdy więcej zawieszeń komputera przy szybkiej transmisji przez modem. Zwolnij porty i odzyskaj część mocy swojej Amigi kupując kartę ioExtender!



ImageFX
Doskonale zintegrowana obróbka grafiki. ImageFX to jedyny pakiet do obróbki grafiki, jakiego będziesz kiedykolwiek potrzebował. To optymalne rozwiązanie dające nie tylko możliwość konwersji pomiędzy różnymi formatami grafiki, jak TIFF, GIF i TARGA, ale również bardzo rozbudowany, 24–bitowy program do malowania i retuszu. Efekty swojej pracy widzisz natychmiast! Wygładzanie brzegów wstawianych elementów, kanał Alpha, operacje ustawiane dla CMY/HSV/YUV/YIQ, skanowanie i digitalizacja, wybór obszaru do obróbki... Wszystko to jest w ImageFX v1.50.

A500
A600
A1200
A2000
A3000
A4000

aram
EXPORT - IMPORT - REKLAMA
Autoryzowany Dealer firmy GVP

ul. Śniadeckich 17
00-654 Warszawa
tel./fax (22) 29-54-99

Skład na komputerze AMIGA 2000 z kartą G-Force'030–50 Combo oraz EGS 28/24 SPECTRUM:
CAT-SOFT

Prowadzimy sprzedaż hurtową, detaliczną oraz wysyłkową za zaliczeniem pocztowym



producent: Rombo
dystrybutor: Rombo, Kirkton Campus, Livingston, Scotland, EH54 7AZ, tel. (44) 0506 414631, fax. (44) 0506 414634
cena: ok. 300 funtów (wersja uboższa 200 funtów)
zgodność: 1.3, 2.0, 3.0

Digitizer w nowej klasie (Vidi Amiga (24) RT)

Chodzi oczywiście o digitizer umożliwiający przetworzenie obrazu telewizyjnego na obraz Amigi. Vidi Amiga jest digitizerem czasu rzeczywistego i może być podłączony do dowolnej Amigi 500 lub 1200. Wystarczy teraz doprowadzić do niego dowolny sygnał video, a nasza księżniczka w dowolnym trybie pokaże nam co z tego wynikło. Digitizer współdziała ze wszystkimi rozdzielczościami i w pełni wykorzystuje możliwości nowych chipów graficznych AGA. Jakość obrazów jest porównywalna do jakości obrazów uzyskiwanych w trakcie skanowania. Jeżeli do takiego cacka bezpośrednio podłączymy kamerę i nie będziemy korzystali z pośrednictwa taśmy, obraz faktycznie będzie idealny. Digitizer wyposażony w odpowiedni program umożliwia zgrywanie plików w wielu popularnych formatach: ILBM, ANIM, BMP, PCX oraz TIFF.

Istnieje możliwość współpracy digitizera z zestawami animacyjnymi tak, że bez większych problemów można dzięki niemu przygotowywać materiały do filmów (niekoniecznie rysunkowych).

Firma oferuje wersję uboższą digitizera oznaczoną jako Vidi Amiga (12) RT. Wersja ta umożliwia "łapanie" grafiki w 4096 kolorach w czasie rzeczywistym, co dla wielu zastosowań domowych jest w zupełności wystarczające.

Każda z wersji posiada wejście dla systemów SVHS oraz Y/C.



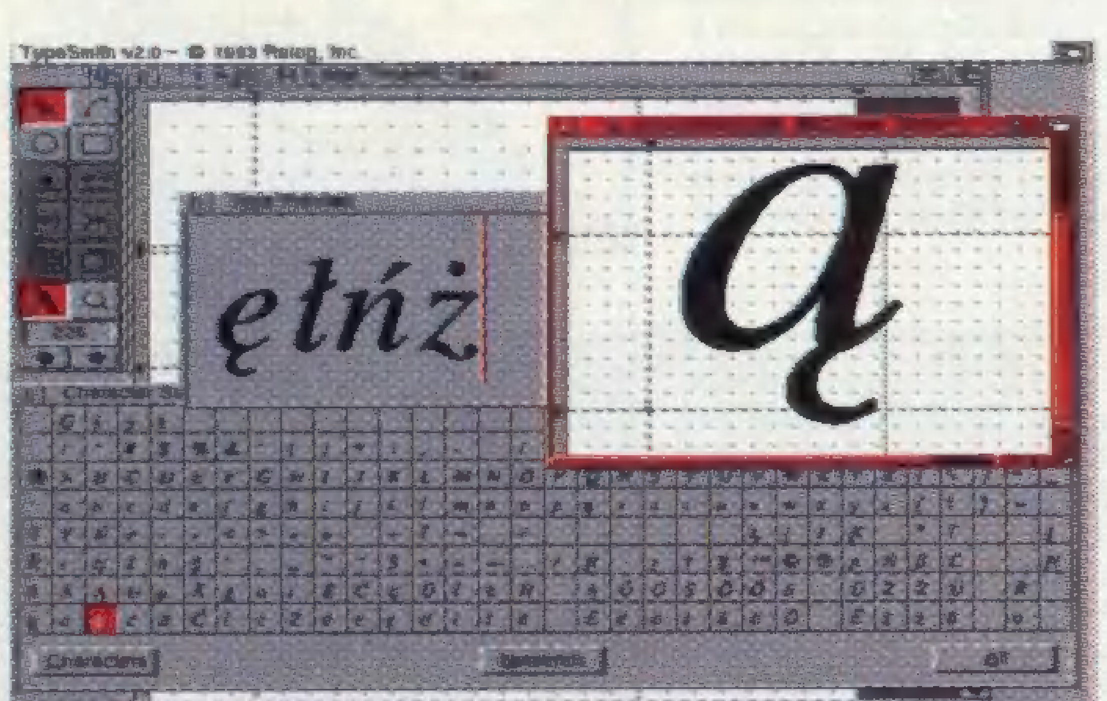
producent: Seikosha Ltd, Unit 14, Poyle 14, Newlands Drive, Coln Brook FL3 ODX tel. 0753 685873
dystrybutor: Zakład Usług Komputerowych "On Line"
cena: ponad 7 mln złotych
zgodność: każda Amiga

Atramentem polane (SpeedJet 200)

Całkiem znośną propozycję przedstawiła firma Seikosha. Drukarka SpeedJet daje wydruki porównywalne pod względem jakości do innych drukarek atramentowych. Jest to model z klasy "niższych", a więc jego cena nie jest zawrotna. Wygodny jest panel obsługi drukarki. Wszelkie opcje jak wielkość papieru, czy rodzaj czcionki ustawia się na specjalnym panelu za pomocą zwykłych przełączników. Drukarka ma jednak i swoje ciemne strony. Wkładanie papieru nie jest zbyt wygodne i trzeba wykonać sporo manipulacji, aby kartka została wciągnięta. Istnieje możliwość ułożenia do 100 kartek na podajniku i problem z głowy.

Drukarka nie należy do najcichszych, ale za to zajmuje na biurku niewiele miejsca. Inną jej wadą jest konieczność ręcznego czyszczenia głowic przy pomocy specjalnego kawałka plastiku, co w innych drukarkach odbywa się automatycznie. Szczególnie warto polecić tę drukarkę małym firmom, które stawiają na wydruki o lepszej jakości bez specjalnego nacisku na ich wielość.

Jeśli chodzi o polskie znaki, to drukarkę wyposażono w zestawy podstawowych czcionek w standardach Latin i Mazovii. Nie wiemy jednak jak wygląda współpraca z Amigą, a niektóre firmy sprzedające drukarkę nie są zainteresowane udzielaniem informacji na ten temat (Optimus, Bydgoszcz: "My się Amigami nie zajmujemy.")



producent: Soft-Logik Publishing
dystrybutor: P.W.H. "Alfin" sp z o.o. ul. Świętojańska 2/7 85-017 Bydgoszcz tel. 28-79-20 fax. 22-64-03
cena: 195$
zgodność: 2.x i wyższe, min. 2MB RAMu

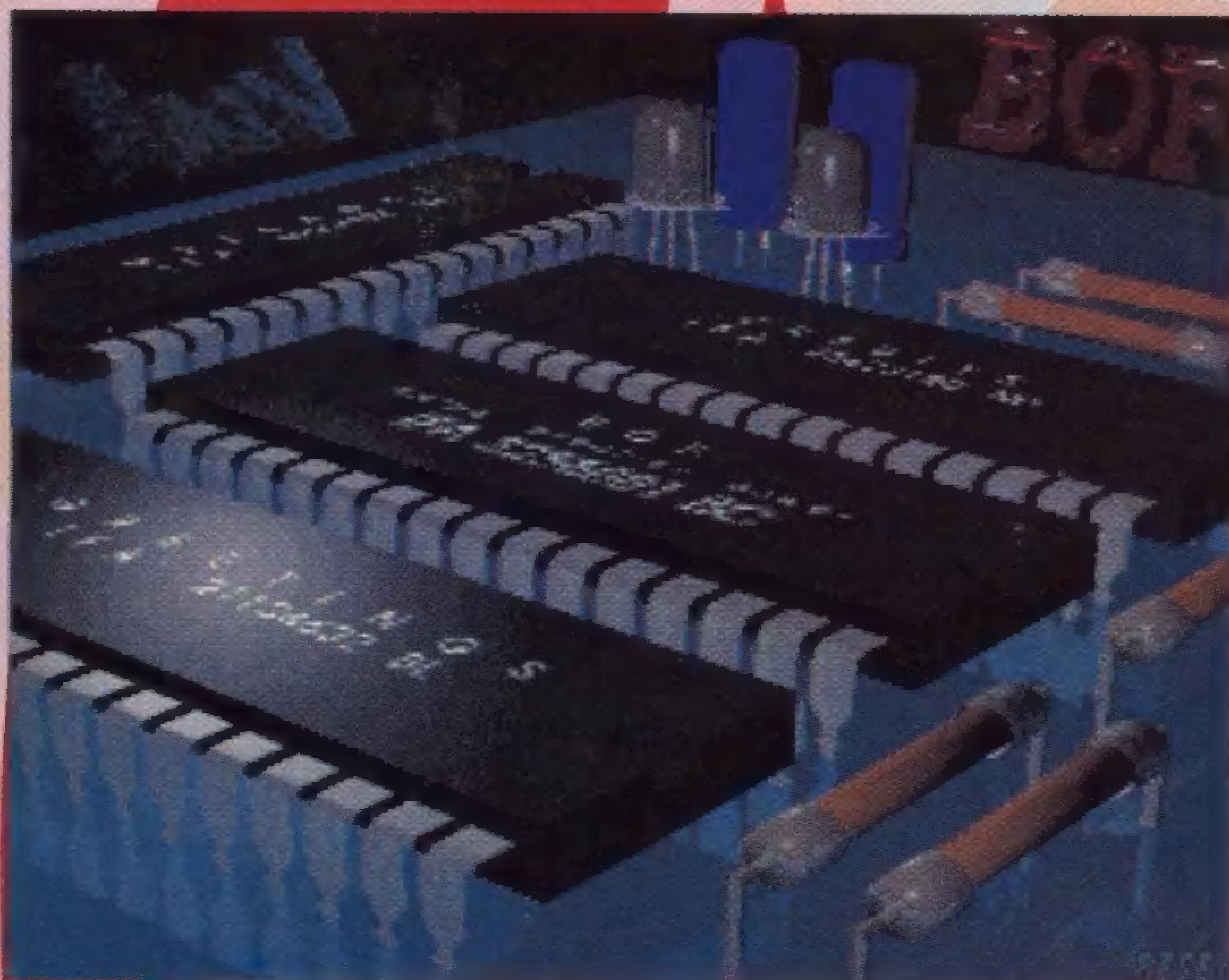
TypeSmith 2.0

Program TypeSmith 2.0 jest przeznaczony dla projektantów krojów czcionek. TypeSmith pozwala na tworzenie czcionek w następujących formatach: PostScript Type 1, PostScript Type 3, Compugraphic Intellifont oraz w formacie Soft-Logika. Tecztery formaty zawierają wszystkie główne standardy stosowane na Amidze. Przy pomocy TypeSmitha można tworzyć czcionki dla takich programów jak: PageStream, Art Expression, Final Copy, Scala i wielu innych.

Program umożliwia zarówno tworzenie nowych czcionek jak i dodawanie własnych znaków do czcionek już istniejących. Nowa wersja zapewnia o wiele wygodniejszą płaszczyznę obsługi. Program stara się zminimalizować niezgodności pomiędzy formatami np. Intellifont i czcionkami PostScriptowymi, poprzez tworzenie np. nowych tabel. Obszerna instrukcja i dodatkowo wygodny w użyciu Help w systemie AmigaGuide pozwalają na szybką orientację w podstawowych funkcjach programu, a także na powolne wgryzanie się w funkcje bardziej fachowe.

Program pozwala na korzystanie z bitmapowych podkładów, tak że stworzenie nowej czcionki wektorowej, dla istniejącego wzorca bitmapowego zajmuje znacznie mniej czasu niż w wielu programach tego typu.

Miło mi donieść, że wkrótce pojawi się polska wersja tego programu.


BOF

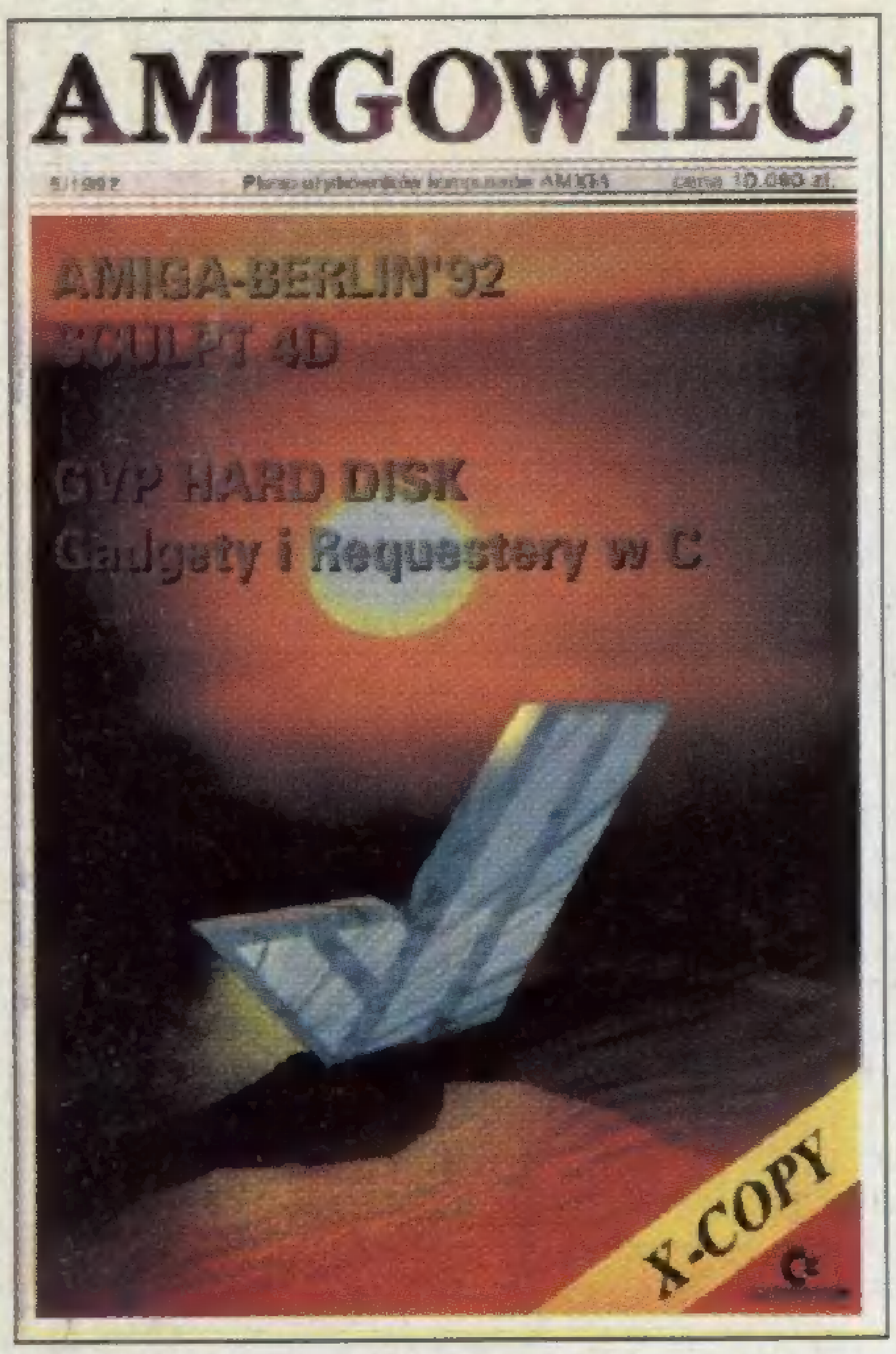


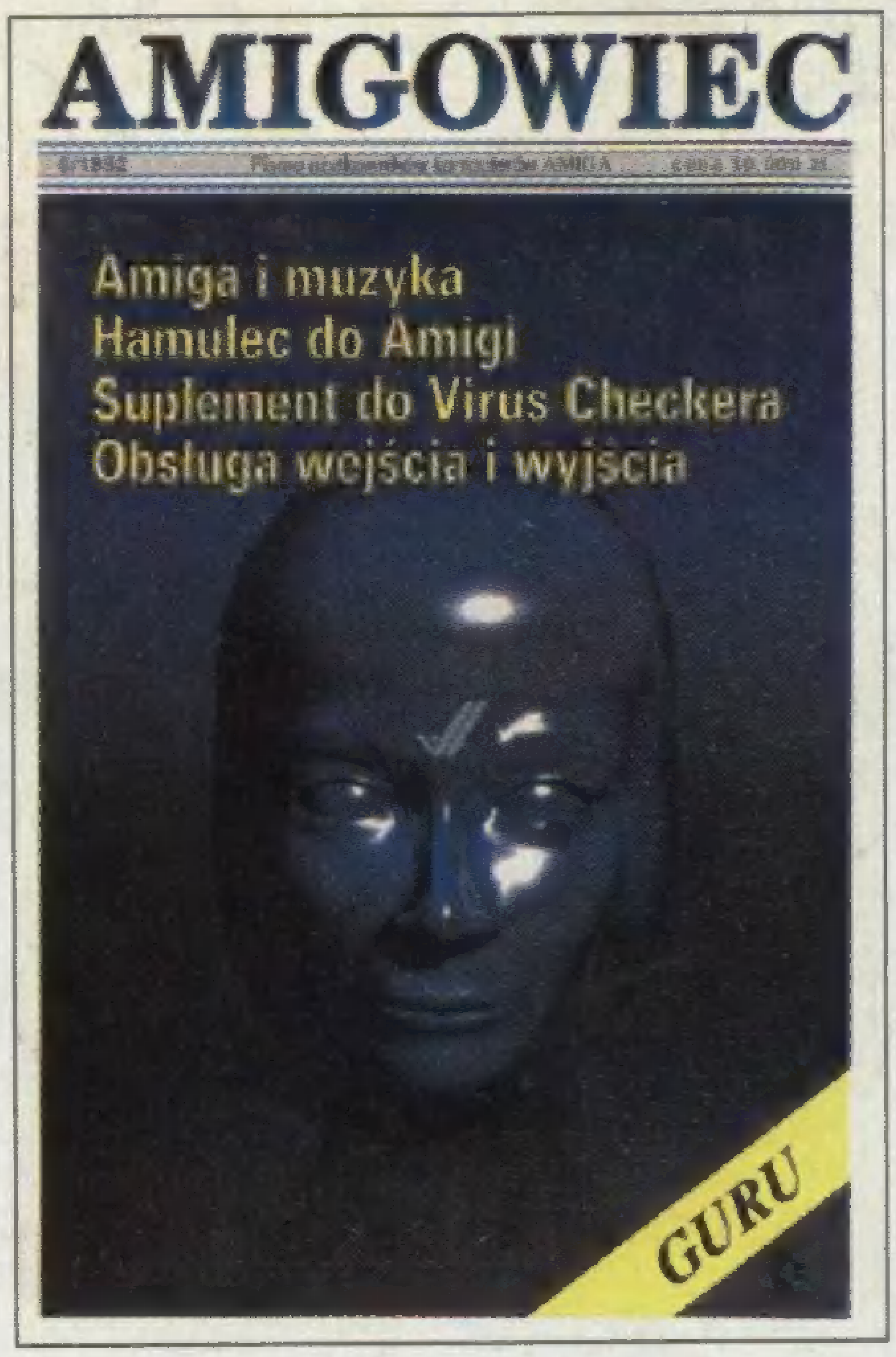




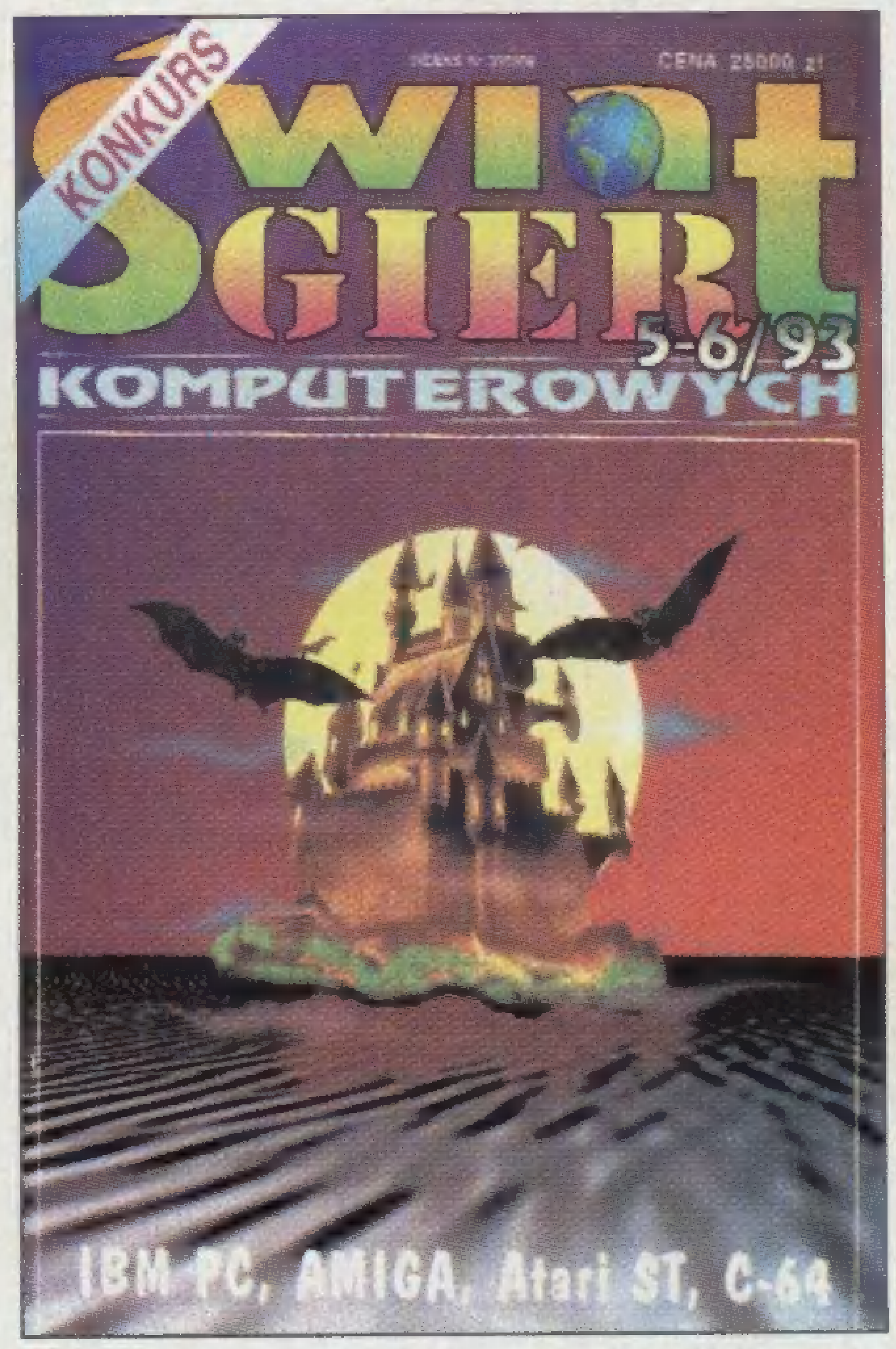
















ZAMÓWIENIE

imię ....

nazwisko ....

adres ....

TEN ZESTAW MOŻE BYĆ

TWÓJ

ZA JEDYNE

80 tys. zł

+ AMIGOWIEC NIESPODZIANKA

Swoje zamówienie wyślij na adres:

ALFIN

sp. z o.o.

ul. Świętojańska 2/7
85-017 Bydgoszcz
tel. 28-79-20
fax 22-64-03